On the computational complexity of cut-reduction

Klaus Aehlig
Computer Science
Swansea University
Swansea SA2 8PP, UK
k.t.aehlig@swansea.ac.uk

Arnold Beckmann
Computer Science
Swansea University
Swansea SA2 8PP, UK
a.beckmann@swansea.ac.uk

February 2, 2008

Abstract

We investigate the complexity of cut-reduction on proof notations, in particular identifying situations where cut-reduction operates feasibly, i.e., sub-exponential, on proof notations. We then apply the machinery to characterise definable search problem in Bounded Arithmetic.

To explain our results with an example, let $\mathbb{E}(d)$ denote Mints' continuous cut-reduction operator which reduces the complexity of all cuts of a propositional derivation d by one level. We will show that if all sub-proofs of d can be denoted with notations of size s, and the height of d is h, then sub-proofs of the derivation $\mathbb{E}(d)$ can be denoted by notations of size $h \cdot (s + \mathcal{O}(1))$. Together with the observation that determining the last inference of a denoted derivation as well as determining notations for immediate sub-derivations is easy (i.e., polynomial time computable), we can apply this result to re-obtain that the Σ^{b}_i-definable functions of the Bounded Arithmetic theory S^i_2 are in the i-th level of the polynomial time hierarchy of functions $\mathrm{FP}^{\Sigma^{\mathrm{b}}_{i-1}}$.

1 Introduction and Related Work

Since Gentzen's invention of the "Logik Kalkül" LK and the proof of his "Hauptsatz" [Gen35a, Gen35b], cut-elimination has been studied in many papers on proof theory. Mints' invention of continuous normalisation [Min78, KMS75] isolates operational aspects of normalisation, that is the manipulations on (infinitary) propositional derivations. These operational aspects are described independently of the system's proof theoretic complexity, but at the expense of introducing the void logical rule of *repetition* to balance derivation trees.

$$\frac{\Gamma}{\Gamma}\,(\mathcal{R})$$

Note that this rule is both logically valid and preserves the sub-formula property, which in particular means that it does not harm computational tasks related to derivations as long as it does not occur too often.

It is well-known that, using $(\mathcal{R})$, the cut-elimination operator becomes a primitive recursive function which is continuous w.r.t. the standard metric on infinitary trees: the normalisation procedure requires only as much information of the input as it produces output, using $(\mathcal{R})$ as the last inference rule of the normal derivation, if the result cannot immediately be determined ("please wait").

In fact, associating some of the repetition rules with computation steps bounds for the simply-typed lambda calculus can be obtained that bound the sum of the number of computation steps and the size of the output [AJ05], strengthening earlier results by Beckmann [Bec01]. Using Schütte's ω-rule [Sch51] this method can also be applied to Gödel's [Göd58] system T.

In this report, we will re-examine this situation. We will show that the cut-reduction operator can be understood as a polynomial time operation natural way, see Observation 9.12. We will work with proof notations which give implicit descriptions of (infinite) propositional proofs: a proof notation system will be a set which is equipped with some functions, most importantly two which compute the following tasks:

- Given a notation h, compute the last inference $\mathrm{tp}(h)$ in the denoted proof.
- Given a notation h and a number $i \in \mathbb{N}$, compute a notation $h[i]$ for the i-th immediate sub-derivation of the derivation denoted by h.

Implicit proof notations given in this way uniquely determine a propositional derivation tree, by exploring the derivation tree from its root and determining the inference at each node of the tree. The cut-reduction operator will be defined on such implicitly described derivation trees. For this, we build on Buchholz' technical very smooth approach to notation systems for continuous cut-elimination [Buc91, Buc97]. Our main result of the first part of the report in particular implies the following statement, as can be seen from Corollary 9.11. Let $2_n(x)$ denote the n-fold iteration of exponentiation 2^x.

> Let d be some propositional derivation, and assume that all sub-proofs of d can be denoted with notations of size bounded by s, and that the height of d is h. Then, all sub-proofs of the derivation obtained from d by reducing the complexity of cut-formulae by k can be denoted by notations of size bounded by $2_{k-1}(2h) \cdot s$.

Observe that the size of notations is exponential only in the height of the original derivation. In the second part of this report we will identify situations occurring in proof-theoretical investigations of Bounded Arithmetic where this height is bounded by an iterated logarithm of some global size parameter, making these sizes feasible.

Bounded Arithmetic has been introduced by Buss [Bus86] as theories of arithmetic with a strong connection to computational complexity. For sake of simplicity of this introduction, we will concentrate only on the Bounded Arithmetic theories S^i_2 by Buss [Bus86]. These theories are given as first order theories of arithmetic in a language which suitably extends that of Peano Arithmetic where induction is restricted in two ways. First, logarithmic induction is considered which only inducts over a logarithmic part of the universe of discourse.

$$\varphi(0) \wedge (\forall x)(\varphi(x) \;\rightarrow\; \varphi(x+1)) \;\rightarrow\; (\forall x)\varphi(|x|) \enspace .$$

Here, $|x|$ denotes the length of the binary representation of the natural number x, which defines a kind of logarithm on natural numbers. Second, the properties which can be inducted on, must be described by a suitably restricted ("bounded") formula. The class of formulae used here are the Σ^{b}_i-formulae which exactly characterise Σ^p_i, that is, properties of the i-th level of the polynomial time hierarchy of predicates. The theory's S^i_2 main ingredients are the instances of logarithmic induction for Σ^{b}_i formulae.

Let a (multi-)function f be called Σ^{b}_j-definable in S^i_2, if its graph can be expressed by a Σ^{b}_j-formula φ, such that the totality of f, which renders as $(\forall x)(\exists y)\varphi(x,y)$, is provable from the S^i_2-axioms in first-order logic. The main results characterising definable (multi-) functions in Bounded Arithmetic are the following.

- Buss [Bus86] has characterised the Σ^{b}_i-definable functions of S^i_2 as $\mathrm{FP}^{\Sigma^b_{i-1}}$, the i-th level of the polynomial time hierarchy of functions.
- Krajíček [Kra93] has characterised the $\Sigma^{\mathrm{b}}_{i+1}$-definable multi-functions of S^i_2 as the class $\mathrm{FP}^{\Sigma^b_i}[wit, \mathcal{O}(\log n)]$ of multi-functions which can be computed in polynomial time using a witness oracle from Σ^p_i, where the number of oracle queries is restricted to $\mathcal{O}(\log n)$ many (n being the length of the input).
- Buss and Krajíček [BK94] have characterised the $\Sigma^{\mathrm{b}}_{i-1}$-definable multi-functions of S^i_2 as projections of solutions to problems from $\mathrm{PLS}^{\Sigma^b_{i-2}}$, which is the class of polynomial local search problems relativised to Σ^p_{i-2}-oracles.

We will re-obtain all these definability characterisations by one unifying method using the results from the first part of this report in the following way. First, we will define a suitable notation system $\mathcal{H}_{\mathrm{BA}}$ for propositional derivations which are obtained by translating Bounded Arithmetic proofs. The propositional translation used here is well-known in proof-theoretic investigations; the translation has been described by Tait [Tai68], and later was independently discovered by Paris and Wilkie [PW85]. In the Bounded-Arithmetic world it is known as the *Paris-Wilkie translation.*

Applying the machinery from the first part we obtain a notation system $\mathcal{CH}_{\mathrm{BA}}$ of cut-elimination for $\mathcal{H}_{\mathrm{BA}}$. $\mathcal{CH}_{\mathrm{BA}}$ will have the property that its implicit

descriptions, most notably the functions $\mathrm{tp}(h)$ and $h[i]$ mentioned above, will be polynomial time computable.

This allows us to formulate a general local search problem on $\mathcal{CH}_{\mathrm{BA}}$ which is suitable to characterise definable multi-functions for Bounded Arithmetic. Assume that $(\forall x)(\exists y)\varphi(x, y)$, describing the totality of some multi-function, is provable in some Bounded Arithmetic theory. Fix a particularly nice formal proof p of this. Given $N \in \mathbb{N}$ we want to describe a procedure which finds some K such that $\varphi(\underline{N}, \underline{K})$ holds. Invert the proof p of $(\forall x)(\exists y)\varphi(x, y)$ to a proof of $(\exists y)\varphi(x, y)$ where x is fresh a variable, then substitute $\underline{N}$ for all occurrences of x. This yields a proof of $(\exists y)\varphi(\underline{N}, y)$. Adding an appropriate number of cut-reduction operators we obtain a proof with all cut-formulae of (at most) the same logical complexity as φ. It should be noted that a notation $h(N)$ for this proof can be computed in time polynomial in N.

The general local search problem which finds a witness for $(\exists y)\varphi(\underline{N}, y)$ can now be characterised as follows. Its instance is given by N. The set of solutions are those notations of a suitable size, which denote a derivation having the property that the derived sequent is equivalent to $(\exists y)\varphi(\underline{N}, y) \vee \psi_1 \vee \cdots \vee \psi_l$ where all ψ_i are "simple enough" and false. An initial solution is given by $h(N)$. A neighbour to a solution h is a solution which denotes an immediate sub-derivation of the derivation denoted by h, if this exists, and h otherwise. The cost of a notation is the height of the denoted derivation. The search task is to find a notation in the set of solutions which is a fixpoint of the neighbourhood function. Obviously, a solution to the search task must exist. In fact, any solution of minimal cost has this property. Now consider any solution to the search problem. It must have the property, that none of the immediate sub-derivations is in the solution space. This can only happen if the last inference derives $(\exists y)\varphi(\underline{N}, y)$ from a true statement $\varphi(\underline{N}, \underline{K})$ for some $K \in \mathbb{N}$. Thus K is a witness to $(\exists y)\varphi(\underline{N}, y)$, and we can output K as a solution to our original witnessing problem.

Depending on the complexity of logarithmic induction present in the Bounded Arithmetic theory we started with, and the level of definability, we obtain local search problems defined by functions of some level of the polynomial time hierarchy, and different bounds to the cost function. For example, if we start with the Σ^{b}_i-definable functions of S^i_2, we obtain a local search problem defined by properties in $\mathrm{FP}^{\Sigma^{\mathrm{b}}_{i-1}}$, where the cost function is bounded by $|N|^{\mathcal{O}(1)}$. Thus, by following the canonical path through the search problem which starts at the initial value and iterates the neighbourhood function, we obtain a path of polynomial length, which describes a procedure in $\mathrm{FP}^{\Sigma^{\mathrm{b}}_{i-1}}$ to compute a witness.

Other research related to our investigations is a paper by Buss [Bus04] which also makes use of the Paris-Wilkie translation to obtain witnessing results by giving uniform descriptions of translated proofs. However, Buss' approach does not explicitely involve cut-elimination. Dynamic ordinal analysis [Bec03, Bec06] characterises the heights of propositional proof trees obtained via the Paris-Wilkie translation and cut-reduction. Therefore, it is not surprising that the

bounds obtained by dynamic ordinal analysis coincide with the bounds on cost functions we are exploiting here.

The potential of our approach to the characterisations of definable search problems via notation systems is that it may lead to characterisations of so far uncharacterised definable search problems, most notably the Σ^{b}_1-definable search problems in S^i_2 for $i \geq 3$.

2 Proof Systems

Let S be a set. The set of all subsets of S will be denoted by $\mathfrak{P}(S)$, the set of all finite subsets of S will be denoted by $\mathfrak{P}_{\mathrm{fin}}(S)$.

Definition 2.1 (sequent)**.** Let $\mathcal{F}$ be a set (of *formulae*), $\approx$ a binary relation on $\mathcal{F}$ (*identity between formulae*), and $\mathrm{rk}\colon \mathfrak{P}(\mathcal{F}) \times \mathcal{F} \to \mathbb{N}$ a function (*rank*). A *sequent* over $\mathcal{F}, \approx, \mathrm{rk}$ is a finite subset of $\mathcal{F}$. We use $\Gamma, \Delta, \dots$ as syntactic variables to denote sequents. With $\approx\!\Delta$ we denote the set $\{A \in \mathcal{F} \colon (\exists B \in \Delta) A \approx B\}$.

We usually write $A_1, \dots, A_n$ for $\{A_1, \dots, A_n\}$ and A, Γ, Δ for $\{A\} \cup \Gamma \cup \Delta$, etc. We always write $\mathcal{C}$-$\mathrm{rk}(A)$ instead of $\mathrm{rk}(\mathcal{C}, A)$.

We repeat standard Buchholz notation for proof systems [Buc97].

Definition 2.2. A *proof system* $\mathfrak{S}$ *over* $\mathcal{F}, \approx, \mathrm{rk}$ is given by

- a set of formal expressions called *inference symbols* (syntactic variable $\mathcal{I}$);
- for each inference symbol $\mathcal{I}$ an ordinal $|\mathcal{I}| \leq \omega$, a sequent $\Delta(\mathcal{I})$ and a family of sequents $(\Delta_\iota(\mathcal{I}))_{\iota < |\mathcal{I}|}$.

Proof systems may have inference symbols of the form Cut_C for $C \in \mathcal{F}$; these are called "cut inference symbols" and their use will (in Definition 2.4) be measured by the $\mathcal{C}$-cut rank.

Notation 2.3. By writing $(\mathcal{I})\,\dfrac{\dots\Delta_\iota\dots(\iota < I)}{\Delta}$ we declare $\mathcal{I}$ as an inference symbol with $|\mathcal{I}| = I$, $\Delta(\mathcal{I}) = \Delta$, $\Delta_\iota(\mathcal{I}) = \Delta_\iota$. If $|\mathcal{I}| = n$ we write $\dfrac{\Delta_0\ \Delta_1\ \dots\ \Delta_{n-1}}{\Delta}$ instead of $\dfrac{\dots\Delta_\iota\dots(\iota < I)}{\Delta}$.

Definition 2.4 (Inductive definition of $\mathfrak{S}$-quasi derivations)**.** If $\mathcal{I}$ is an inference symbol of $\mathfrak{S}$, and $(d_\iota)_{\iota<|\mathcal{I}|}$ is a sequence of $\mathfrak{S}$-quasi derivations, then $d :=$

$\mathcal{I}(d_\iota)_{\iota<|\mathcal{I}|}$ is an $\mathfrak{S}$-*quasi derivation* with

$$
\begin{aligned}
\Gamma(d) &:= \Delta(\mathcal{I}) \cup \bigcup_{\iota<|\mathcal{I}|} (\Gamma(d_\iota) \setminus {\approx}\Delta_\iota(\mathcal{I})) && (\textit{endsequent of } d)\\
\mathrm{last}(d) &:= \mathcal{I} && (\textit{last inference of } d)\\
d(\iota) &:= d_\iota \text{ for } \iota < |\mathcal{I}| && (\textit{sub-derivation})\\
\mathcal{C}\text{-crk}(d) &:= \sup(\{\mathcal{C}\text{-rk}(\mathcal{I})\} \cup \{\mathcal{C}\text{-crk}(d_\iota) \colon \iota < |\mathcal{I}|\}) && (\textit{cut-rank of } d)\\
&\qquad \text{where } \mathcal{C}\text{-rk}(\mathcal{I}) := \begin{cases} \mathcal{C}\text{-rk}(C)+1 & \text{if } \mathcal{I} = \mathrm{Cut}_C \\ 0 & \text{otherwise} \end{cases}\\
\mathrm{hgt}(d) &:= \sup\{\mathrm{hgt}(d_\iota)+1 \colon \iota < |\mathcal{I}|\} && (\textit{height of } d)\\
\mathrm{sz}(d) &:= (\sum_{\iota<|\mathcal{I}|} \mathrm{sz}(d_\iota)) + 1 && (\textit{size of } d)
\end{aligned}
$$

3 The infinitary proof system

Definition 3.1. Let $\mathbb{C} = \{\top, \bot, \bigwedge, \bigvee\}$ be the set of (symbols for) connectives for infinitary logic. Their arity is given by $|\top| = |\bot| = 0$ and $|\bigwedge| = |\bigvee| = \omega$. We define a negation of the connectives according to the de Morgan laws: $\neg(\top) = \bot$, $\neg(\bot) = \top$, $\neg(\bigwedge) = \bigvee$, and $\neg(\bigvee) = \bigwedge$.

Definition 3.2. The set of all infinitary formulae $\mathcal{L}_\infty$ together with their rank is inductively defined by the clause: if $c \in \mathbb{C}$ and $A_\iota \in \mathcal{L}_\infty$ for $\iota < |c|$ then $c(A_\iota)_{\iota<|c|} \in \mathcal{L}_\infty$ and $\mathcal{C}\text{-rk}(c(A_\iota)_{\iota<|c|}) = \sup_{\iota<|c|}(\mathcal{C}\text{-rk}(A_\iota) + 1)$.

Notation
We denote $\top()$ by $\top$ and $\bot()$ by $\bot$.

Definition 3.3. $\neg$ denotes the operation on $\mathcal{L}_\infty$ which computes negation according to the de Morgan rules, i.e.

$$\neg\left(c(A_\iota)_{\iota<|c|}\right) := \neg(c)\big(\neg(A_\iota)\big)_{\iota<|c|}$$

Definition 3.4. The set of all infinitary formulae of finite rank is denoted with $\mathcal{F}_\infty$. The identity between $\mathcal{F}_\infty$-formulae is the "true" set-theoretic equality.

Definition 3.5. The *infinitary proof system* $\mathfrak{S}_\infty$ is the proof system over $\mathcal{F}_\infty$ which is given by the following set of inference symbols:

(Ax) $\quad \dfrac{}{\top}$

$(\bigwedge_A)$ $\quad \dfrac{\ldots \quad A_\iota \quad \ldots \quad (\iota < \omega)}{A}$ for $A = \bigwedge(A_\iota)_{\iota<\omega} \in \mathcal{F}_\infty$

$(\bigvee_A^i)$ $\quad \dfrac{A_i}{A}$ for $A = \bigvee(A_\iota)_{\iota<\omega} \in \mathcal{F}_\infty$ and $i < \omega$

$$(\mathrm{Cut}_C) \quad \frac{C \qquad \neg C}{\emptyset} \quad \text{for } C \in \mathcal{F}_\infty$$

$$(\mathrm{Rep}) \quad \frac{\emptyset}{\emptyset}$$

Definition 3.6. The $\mathfrak{S}_\infty$-*derivations* are the $\mathfrak{S}_\infty$-quasi derivations.

With a $\mathfrak{S}_\infty$-derivation $d = \mathcal{I}(d_\iota)_{\iota<|\mathcal{I}|}$ we can associate a function from $\mathbb{N}^{<\omega}$ to $\mathfrak{S}_\infty$ by letting $d(\langle\rangle) := \mathrm{last}(d)$ and

$$d(\langle i \rangle \frown s) := \begin{cases} d_i(s) & \text{if } i < |\mathcal{I}| \\ \mathrm{Ax} & \text{otherwise} \end{cases}$$

4 Notation system for infinitary formulae

Definition 4.1. A *notation system for (infinitary) formulae* is a set $\mathcal{F}$ of "formulae", together with four functions $\mathrm{tp}\colon \mathcal{F} \to \{\top, \bot, \bigwedge, \bigvee\}$, $\cdot[\cdot]\colon \mathcal{F} \times \mathbb{N} \to \mathcal{F}$, $\neg\colon \mathcal{F} \to \mathcal{F}$, and $\mathrm{rk}\colon \mathfrak{P}(\mathcal{F}) \times \mathcal{F} \to \mathbb{N}$ called "outermost connective", "subformula", "negation" and "rank", and a relation $\approx \subseteq \mathcal{F} \times \mathcal{F}$ called "intensional equality", such that $\mathrm{tp}(\neg(f)) = \neg(\mathrm{tp}(f))$, $\neg(f)[n] = \neg(f[n])$, $\mathcal{C}\text{-rk}(f) = \mathcal{C}\text{-rk}(\neg f)$, $\mathcal{C}\text{-rk}(f[n]) < \mathcal{C}\text{-rk}(f)$ for $n < |\,\mathrm{tp}(f)|$, and $f \approx g$ implies $\mathrm{tp}(f) = \mathrm{tp}(g)$, $f[n] \approx g[n]$, $\neg(f) \approx \neg(g)$ and $\mathcal{C}\text{-rk}(f) = \mathcal{C}\text{-rk}(g)$.

It should be noted that if $\mathcal{F}$ is a notation system for formulae, then so is $\mathcal{F}/\approx$ in the obvious way; moreover, in $\mathcal{F}/\approx$ the intensional equality is true equality in the quotient. The reason why we nevertheless explicitly consider an (intensional) equality relation is that we are interested in the computational complexity of notation systems and therefore prefer to take notations as the strings that arise naturally, rather than working on the quotient. Note that the latter would require us to compute canonical representations anyway and so would just push the problem to a different place.

It should also be noted that the intensional equality is truly intensional. Two formulae are only equal, if they are given to us as being equal. The obvious extensional equality would be the largest bisimulation, that is, the largest relation $\sim \subset \mathcal{F} \times \mathcal{F}$ satisfying $f \sim g \to \mathrm{tp}(f) = \mathrm{tp}(g) \wedge f[n] \sim g[n] \wedge \mathcal{C}\text{-rk}(f) = \mathcal{C}\text{-rk}(g) \wedge \neg f \sim \neg g$. However, as most extensional concepts, the largest bisimulation is undecidable in almost all interesting cases and therefore not suited for an investigation of effective notations.

Definition 4.2. Let $\mathcal{F} = (\mathcal{F}, \mathrm{tp}, \cdot[\cdot], \mathrm{rk}, \approx)$ be a notation system for infinitary formulae. The *interpretation* $[\![f]\!]_\infty$ *of* $f \in \mathcal{F}$ is inductively defined as

$$[\![f]\!]_\infty = \mathrm{tp}(f)([\![f[\iota]]\!]_\infty)_{\iota<|\,\mathrm{tp}(f)|}$$

Observation 4.3. *The following properties hold.*

1. $f \sim g \quad \Leftrightarrow \quad [\![f]\!]_\infty = [\![g]\!]_\infty$,
2. $f \approx g \quad \Rightarrow \quad [\![f]\!]_\infty = [\![g]\!]_\infty$.

5 Semiformal proof systems

Let $\mathcal{F} = (\mathcal{F}, \mathrm{tp}, \cdot[\cdot], \mathrm{rk}, \approx)$ be a notation system for infinitary formulae.

Definition 5.1. The *semiformal proof system* $\mathfrak{S}_{\mathcal{F}}$ *over* $\mathcal{F}$ is the proof system over $\mathcal{F}$ which is given by the following set of inference symbols:

(Ax_A) $\quad \dfrac{}{A}$ for $A \in \mathcal{F}$ with $\mathrm{tp}(A) = \top$

$(\bigwedge_C)$ $\quad \dfrac{\ldots \quad C[n] \quad \ldots \quad (n \in \mathbb{N})}{C}$ for $C \in \mathcal{F}$ with $\mathrm{tp}(C) = \bigwedge$

$(\bigvee_C^i)$ $\quad \dfrac{C[i]}{C}$ for $C \in \mathcal{F}$ with $\mathrm{tp}(C) = \bigvee$ and $i \in \mathbb{N}$

(Cut_C) $\quad \dfrac{C \qquad \neg C}{\emptyset}$ for $C \in \mathcal{F}$ with $\mathrm{tp}(C) \in \{\top, \bigwedge\}$

(Rep) $\quad \dfrac{\emptyset}{\emptyset}$

Abbreviations

For $\mathrm{tp}(C) \in \{\bot, \bigvee\}$ let (Cut_C) $\dfrac{C \qquad \neg C}{\emptyset}$ denote $(\mathrm{Cut}_{\neg C})$ $\dfrac{\neg C \qquad C}{\emptyset}$.

Definition 5.2. The $\mathfrak{S}_{\mathcal{F}}$*-derivations* are the $\mathfrak{S}_{\mathcal{F}}$-quasi derivations.

Later in our applications, we will be concerned only with derivations of finite height, for which we can formulate slightly sharper upper bounds on cut-reduction than in the general (infinite) case (2^α versus 3^α). Thus, from now on we will restrict attention to derivations of finite height only.

Definition 5.3. Let $d \vdash^{\alpha}_{\mathcal{C},m} \Gamma$ denote that d is an $\mathfrak{S}_{\mathcal{F}}$-derivation with $\Gamma(d) \subseteq\ \approx\!\Gamma$, $\mathcal{C}$-$\mathrm{crk}(d) \leq m$, and $\mathrm{hgt}(d) \leq \alpha < \omega$.

Definition 5.4. The *interpretation* $[\![d]\!]_\infty$ *of a* $\mathfrak{S}_{\mathcal{F}}$*-derivation* $d = \mathcal{I}(d_\iota)_{\iota<|\mathcal{I}|}$ is defined as

$$[\![d]\!]_\infty := [\![\mathcal{I}]\!]_\infty([\![d_\iota]\!]_\infty)_{\iota<|\mathcal{I}|}$$

where $[\![\mathcal{I}]\!]_\infty$ is defined by

$$\begin{aligned} [\![\mathrm{Ax}_A]\!]_\infty &:= \mathrm{Ax} \\ [\![\textstyle\bigwedge_A]\!]_\infty &:= \textstyle\bigwedge_{[\![A]\!]_\infty} \\ [\![\textstyle\bigvee_A^i]\!]_\infty &:= \textstyle\bigvee^i_{[\![A]\!]_\infty} \\ [\![\mathrm{Cut}_C]\!]_\infty &:= \mathrm{Cut}_{[\![C]\!]_\infty} \\ [\![\mathrm{Rep}]\!]_\infty &:= \mathrm{Rep} \end{aligned}$$

Observation 5.5. $\Gamma([\![d]\!]_\infty) \subseteq [\![\Gamma(d)]\!]_\infty$

Proof. Induction on d. The "$\subseteq$", instead of the expected "$=$" is due to the fact, that only formulae are removed from the conclusion that are intensionally equal; compare also Observation 4.3. ∎

6 Cut elimination for semiformal systems

Let $\mathcal{F} = (\mathcal{F}, \mathrm{tp}, \cdot[\cdot], \mathrm{rk}, \approx)$ be a notation system for infinitary formulae, and $\mathfrak{S}_\mathcal{F}$ the semiformal proof system over $\mathcal{F}$. We define Mints' continuous cut-reduction operator [Min78, KMS75] following the description given by Buchholz [Buc91]. The only modification is our explicit use of intensional equality.

Theorem 6.1 (and Definition)**.** *Let $C \in \mathcal{F}$ with $\mathrm{tp}(C) = \bigwedge$, and $k < \omega$ be given. We define an operator $\mathbb{I}_C^k$ such that:* $\quad d \vdash^\alpha_{\mathcal{C},m} \Gamma, C \quad \Rightarrow \quad \mathbb{I}_C^k(d) \vdash^\alpha_{\mathcal{C},m} \Gamma, C[k]$.

Proof by induction on the build-up of d: W.l.o.g. we may assume that $\Gamma = \Gamma(d) \setminus \approx\{C\}$.

Case 1. $\mathrm{last}(d) \in \{\bigwedge_D : D \approx C\}$. Then

$$\mathbb{I}_C^k(d) := \mathrm{Rep}(\mathbb{I}_C^k(d(k)))$$

is a derivation as required.

Case 2. $\mathcal{I} := \mathrm{last}(d) \notin \{\bigwedge_D : D \approx C\}$. Then

$$\mathbb{I}_C^k(d) := \mathcal{I}(\mathbb{I}_C^k(d(i)))_{i<|\mathcal{I}|}$$

is a derivation as required. ∎

Theorem 6.2 (and Definition)**.** *Let $C \in \mathcal{F}$ with $\mathrm{tp}(C) \in \{\top, \bigwedge\}$ be given. We define an operator $\mathbb{R}_C$ such that:* $\quad d_0 \vdash^\alpha_{\mathcal{C},m} \Gamma, C \quad \& \quad d_1 \vdash^\beta_{\mathcal{C},m} \Gamma, \neg C \quad \& \quad \mathcal{C}\text{-}\mathrm{rk}(C) \le m \quad \Rightarrow \quad \mathbb{R}_C(d_0, d_1) \vdash^{\alpha+\beta}_{\mathcal{C},m} \Gamma$.

Proof by induction on the build-up d: W.l.o.g. we may assume that $\Gamma = (\Gamma(d_0) \setminus \approx\{C\}) \cup (\Gamma(d_1) \setminus \approx\{\neg C\})$. Let $\mathcal{I} = \mathrm{last}(d_1)$.

Case 1. $\Delta(\mathcal{I}) \cap \approx\{\neg C\} = \emptyset$. Then $\Delta(\mathcal{I}) \subseteq \Gamma$ and $d_1(i) \vdash^{\beta_i}_{\mathcal{C},m} \Gamma, \neg C, \Delta_i(\mathcal{I})$ with $\beta_i < \beta$ for all $i < |\mathcal{I}|$. By induction hypothesis we obtain $\mathbb{R}_C(d_0, d_1(i)) \vdash^{\alpha+\beta_i}_{\mathcal{C},m} \Gamma, \Delta_i(\mathcal{I})$ for $i < |\mathcal{I}|$. Hence

$$\mathbb{R}_C(d_0, d_1) := \mathcal{I}(\mathbb{R}_C(d_0, d_1(i)))_{i<|\mathcal{I}|}$$

is a derivation as required.

Case 2. $\Delta(\mathcal{I}) \cap \approx\{\neg C\} \neq \emptyset$. Then $\mathrm{tp}(C) \neq \top$, because otherwise there is some $D \in \Delta(\mathcal{I})$ with $\mathrm{tp}(D) = \bot$, but this is not satisfied by any of the inference symbols of the semiformal system $\mathfrak{S}_\mathcal{F}$. Hence $\mathrm{tp}(C) = \bigwedge$. We obtain that $\mathcal{I} = \bigvee_D^k$ for some $k \in \mathbb{N}$ and $D \approx \neg C$, and $d_1(0) \vdash^{\beta_0}_{\mathcal{C},m} \Gamma, \neg C, \neg C[k]$ with $\beta_0 < \beta$. By induction hypothesis we obtain $\mathbb{R}_C(d_0, d_1(0)) \vdash^{\alpha+\beta_0}_{\mathcal{C},m} \Gamma, \neg C[k]$. The Inversion Theorem shows $\mathbb{I}_C^k(d_0) \vdash^\alpha_{\mathcal{C},m} \Gamma, C[k]$. Now $\mathcal{C}\text{-}\mathrm{rk}(C[k]) < \mathcal{C}\text{-}\mathrm{rk}(C) \le m$, hence

$$\mathbb{R}_C(d_0, d_1) := \mathrm{Cut}_{C[k]}(\mathbb{I}_C^k(d_0), \mathbb{R}_C(d_0, d_1(0)))$$

is a derivation as required. ∎

Theorem 6.3 (and Definition)**.** *We define an operator* $\mathbb{E}$ *such that:*
$d \vdash^{\alpha}_{\mathcal{C},m+1} \Gamma \quad \Rightarrow \quad \mathbb{E}(d) \vdash^{2^{\alpha}-1}_{\mathcal{C},m} \Gamma.$

Proof by induction on the build-up of d*:* W.l.o.g. we may assume that $\Gamma = \Gamma(d)$.

Case 1. $\mathrm{last}(d) = \mathrm{Cut}_C$. Then $\mathcal{C}\text{-rk}(C) \leq m$ and $d(0) \vdash^{\alpha_0}_{\mathcal{C},m+1} \Gamma, C$ and $d(1) \vdash^{\alpha_0}_{\mathcal{C},m+1} \Gamma, \neg C$ with $\alpha_0 < \alpha$. By induction hypothesis we obtain $\mathbb{E}(d(0)) \vdash^{2^{\alpha_0}-1}_{\mathcal{C},m} \Gamma, C$ and $\mathbb{E}(d(1)) \vdash^{2^{\alpha_0}-1}_{\mathcal{C},m} \Gamma, \neg C$.

Case 1.1. $\mathrm{tp}(C) \in \{\top, \bigwedge\}$, then by the last Theorem $\mathbb{R}_C(\mathbb{E}(d(0)), \mathbb{E}(d(1))) \vdash^{2 \cdot 2^{\alpha_0}-2}_{\mathcal{C},m} \Gamma$, and

$$\mathbb{E}(d) := \mathrm{Rep}(\mathbb{R}_C(\mathbb{E}(d(0)), \mathbb{E}(d(1))))$$

is a derivation as required.

Case 1.2. $\mathrm{tp}(C) \notin \{\top, \bigwedge\}$, then $\mathbb{R}_{\neg C}(\mathbb{E}(d(1)), \mathbb{E}(d(0))) \vdash^{2 \cdot 2^{\alpha_0}-1}_{\mathcal{C},m} \Gamma$. Continue as before.

Case 2. $\mathcal{I} := \mathrm{last}(d) \neq \mathrm{Cut}_C$. Then

$$\mathbb{E}(d) := \mathcal{I}(\mathbb{E}(d(i)))_{i < |\mathcal{I}|}$$

is as required. ∎

Remark 6.4. Immediately from the definition we note that the operators $\mathbb{I}$, $\mathbb{R}$, and $\mathbb{E}$ only inspects the last inference symbol of a derivation to obtain the last inference symbol of the transformed derivation. It should be noted that this continuity would not be possible without the repetition rule.

7 Notations for derivations and cut-elimination

Let $\mathcal{F}$ be a notation system for formulae, and $\mathfrak{S}_{\mathcal{F}}$ the semiformal proof system over $\mathcal{F}$ from Definition 5.1.

Definition 7.1. A *notation system for* $\mathfrak{S}_{\mathcal{F}}$ is a set $\mathcal{H}$ of *notations* and functions $\mathrm{tp}\colon \mathcal{H} \to \mathfrak{S}_{\mathcal{F}}$, $\cdot[\cdot]\colon \mathcal{H} \times \mathbb{N} \to \mathcal{H}$, $\Gamma\colon \mathcal{H} \to \mathfrak{P}_{\mathrm{fin}}(\mathcal{F})$, $\mathrm{crk}\colon \mathfrak{P}(\mathcal{F}) \times \mathcal{H} \to \mathbb{N}$, and $\mathrm{o}, |\cdot|\colon \mathcal{H} \to \mathbb{N} \setminus \{0\}$ called *denoted last inference*, *denoted sub-derivation*, *denoted end-sequent*, *denoted cut-rank*, *denoted height* and *size*, such that $\mathcal{C}\text{-crk}(h[n]) \leq \mathcal{C}\text{-crk}(h)$, $\mathrm{tp}(h) = \mathrm{Cut}_C$ implies $\mathcal{C}\text{-rk}(C) < \mathcal{C}\text{-crk}(h)$, $\mathrm{o}(h[n]) < \mathrm{o}(h)$ for $n < |\,\mathrm{tp}(h)|$, and the following local faithfulness property holds for $h \in \mathcal{H}$:

$$\Delta(\mathrm{tp}(h)) \cup \bigcup_{\iota < |\,\mathrm{tp}(h)|} \Big(\Gamma(h[\iota]) \setminus {\approx}\Delta_\iota(\mathrm{tp}(h))\Big) \subseteq {\approx}\Gamma(h) \ .$$

Proposition 7.2.

$$\Gamma(h[j]) \subseteq {\approx}\Big(\Gamma(h) \cup \Delta_j(\mathrm{tp}(h))\Big)$$

Definition 7.3. Let $\mathcal{H} = (\mathcal{H}, \mathrm{tp}, \cdot[\cdot], \mathrm{o}, |\cdot|)$ be a notation system for $\mathfrak{S}_{\mathcal{F}}$. The *interpretation* $[\![h]\!]$ *of* $h \in \mathcal{H}$ is inductively defined as the following $\mathfrak{S}_{\mathcal{F}}$-derivation:

$$[\![h]\!] := \mathrm{tp}(h)([\![h[n]]\!])_{n<|\,\mathrm{tp}(h)|}$$

Observation 7.4. *For* $h \in \mathcal{H}$ *we have*

$$\begin{aligned} \mathrm{last}([\![h]\!]) &= \mathrm{tp}(h) \\ [\![h]\!](\iota) &= [\![h[\iota]]\!] \qquad \textit{for } \iota < |\,\mathrm{tp}(h)| \\ \Gamma([\![h]\!]) &\subseteq {\approx}\Gamma(h) \end{aligned}$$

We now extend a notation system $\mathcal{H}$ for $\mathfrak{S}_{\mathcal{F}}$ to notation system for cut-elimination on $\mathcal{H}$, by adding notations for the operators $\mathbb{I}$, $\mathbb{R}$ and $\mathbb{E}$ from the previous section.

Definition 7.5. The *notation system* $\mathcal{CH}$ *for cut-elimination on* $\mathcal{H}$ is given by the set of terms $\mathcal{CH}$ which are inductively defined by

- $\mathcal{H} \subset \mathcal{CH}$,
- $h \in \mathcal{CH}$, $C \in \mathcal{F}$ with $\mathrm{tp}(C) = \bigwedge$, $k < \omega \quad \Rightarrow \quad \mathsf{I}^k_C h \in \mathcal{CH}$,
- $h_0, h_1 \in \mathcal{CH}$, $C \in \mathcal{F}$ with $\mathrm{tp}(C) \in \{\top, \bigwedge\} \quad \Rightarrow \quad \mathsf{R}_C h_0 h_1 \in \mathcal{CH}$,
- $h \in \mathcal{CH} \quad \Rightarrow \quad \mathsf{E}h \in \mathcal{CH}$,

where $\mathsf{I}, \mathsf{R}, \mathsf{E}$ are new symbols, and functions $\mathrm{tp}\colon \mathcal{CH} \to \mathfrak{S}_{\mathcal{F}}$, $\cdot[\cdot]\colon \mathcal{CH} \times \mathbb{N} \to \mathcal{CH}$, $\Gamma\colon \mathcal{CH} \to \mathfrak{P}_{\mathrm{fin}}(\mathcal{F})$, $\mathrm{crk}\colon \mathfrak{P}(\mathcal{F}) \times \mathcal{CH} \to \mathbb{N}$, $\mathrm{o}\colon \mathcal{CH} \to \mathbb{N} \setminus \{0\}$ and $|\cdot|\colon \mathcal{CH} \to \mathbb{N}$ defined by recursion on the build-up of $h \in \mathcal{CH}$:

- If $h \in \mathcal{H}$ then all functions are inherited from $\mathcal{H}$.
- $h = \mathsf{I}^k_C h_0$: Let $\Gamma(h) := \{C[k]\} \cup (\Gamma(h_0) \setminus {\approx}\{C\})$, $\mathcal{C}\text{-crk}(h) := \mathcal{C}\text{-crk}(h_0)$, $\mathrm{o}(h) := \mathrm{o}(h_0)$, and $|h| := |h_0| + 1$.

  Case 1. $\mathrm{tp}(h_0) \in \{\bigwedge_D \colon D \approx C\}$. Then let $\mathrm{tp}(h) := \mathrm{Rep}$, and $h[0] := \mathsf{I}^k_C h_0[k]$.

  Case 2. Otherwise, let $\mathrm{tp}(h) := \mathrm{tp}(h_0)$, and $h[i] := \mathsf{I}^k_C h_0[i]$.

- $h = \mathsf{R}_C h_0 h_1$: Let $\mathcal{I} := \mathrm{tp}(h_1)$. We define $\Gamma(h) := (\Gamma(h_0) \setminus {\approx}\{C\}) \cup (\Gamma(h_1) \setminus {\approx}\{\neg C\})$, $\mathcal{C}\text{-crk}(h) := \max\{\mathcal{C}\text{-crk}(h_0), \mathcal{C}\text{-crk}(h_1)\}$, $\mathrm{o}(h) := \mathrm{o}(h_0) + \mathrm{o}(h_1)$, and $|h| := |h_0| + |h_1| + 1$.

  Case 1. $\Delta(\mathcal{I}) \cap {\approx}\{\neg C\} = \emptyset$: Then let $\mathrm{tp}(h) := \mathcal{I}$, and $h[i] := \mathsf{R}_C h_0 h_1[i]$.

  Case 2. Otherwise, $\mathrm{tp}(C) \neq \top$, because if not there would be some $D \in \Delta(\mathcal{I})$ with $\mathrm{tp}(D) = \bot$, but this is not satisfied by any of the inference symbols of the semiformal system $\mathfrak{S}_{\mathcal{F}}$. Hence $\mathrm{tp}(C) = \bigwedge$. Thus $\mathcal{I} = \bigvee^k_D$ for some $k \in \mathbb{N}$ and $D \approx \neg C$. Then let $\mathrm{tp}(h) := \mathrm{Cut}_{C[k]}$ and $h[0] := \mathsf{I}^k_C h_0$, $h[1] := \mathsf{R}_C h_0 h_1[0]$.

- $h = \mathsf{E}h_0$: Let $\Gamma(h) := \Gamma(h_0)$, $\mathcal{C}\text{-crk}(h) := \mathcal{C}\text{-crk}(h_0) \dot{-} 1$, $\mathrm{o}(h) := 2^{\mathrm{o}(h_0)} - 1$, and $|h| := |h_0| + 1$.

  Case 1. $\mathrm{tp}(h_0) = \mathrm{Cut}_C$: Then let $\mathrm{tp}(h) := \mathrm{Rep}$ and
  let $h[0] := \mathsf{R}_C \mathsf{E}h_0[0]\mathsf{E}h_0[1]$ if $\mathrm{tp}(C) \in \{\top, \bigwedge\}$,
  let $h[0] := \mathsf{R}_{\neg C} \mathsf{E}h_0[1]\mathsf{E}h_0[0]$ if $\mathrm{tp}(C) \notin \{\top, \bigwedge\}$.

  Case 2. Otherwise, let $\mathrm{tp}(h) := \mathrm{tp}(h_0)$, and $h[i] := \mathsf{E}h_0[i]$.

Proof. The just defined system is a notation system for $\mathfrak{S}_{\mathcal{F}}$ in the sense of Definition 7.1. To prove this we have to show that

$$\mathrm{o}(h[n]) < \mathrm{o}(h) \qquad \text{for} \qquad n < |\,\mathrm{tp}(h)| \tag{1}$$

and that the local faithfulness property for Γ holds. We start by proving (1) by induction on the build-up of $h \in \mathcal{CH}$.

If $h \in \mathcal{H}$ then (1) is inherited from $\mathcal{H}$. If $h = \mathsf{I}^k_C h_0$ then $h[n] = \mathsf{I}^k_C h_0[n']$ for some n' and (1) is immediate by induction hypothesis.

Now let us consider the case $h = \mathsf{R}_C h_0 h_1$. If $h[n] = \mathsf{R}_C h_0 h_1[n']$ for some n' then (1) is immediate by induction hypothesis. The other case is that $h[0] = \mathsf{I}^k_C h_0$ for some k. We compute

$$\mathrm{o}(h[0]) = \mathrm{o}(\mathsf{I}^k_C h_0) = \mathrm{o}(h_0) < \mathrm{o}(h_0) + \mathrm{o}(h_1) = \mathrm{o}(h)$$

since $\mathrm{o}(h_1) > 0$.

Finally, let us consider the case $h = \mathsf{E}h_0$. If $h[n] = \mathsf{E}h_0[n]$ then (1) is immediate by induction hypothesis. Otherwise, we are in the case $h[0] = \mathsf{R}_C(\mathsf{E}h_0[i])(\mathsf{E}h_0[j])$ for some C, i, j. By induction hypothesis we obtain that $\mathrm{o}(h_0[i]) \le \mathrm{o}(h_0) - 1$ and $\mathrm{o}(h_0[j]) \le \mathrm{o}(h_0) - 1$. Hence

$$\begin{aligned}\mathrm{o}(\mathsf{R}_C(\mathsf{E}h_0[i])(\mathsf{E}h_0[j])) &= \mathrm{o}(\mathsf{E}h_0[i]) + \mathrm{o}(\mathsf{E}h_0[j]) = 2^{\mathrm{o}(h_0[i])} - 1 + 2^{\mathrm{o}(h_0[j])} - 1 \\ &< 2 \cdot 2^{\mathrm{o}(h_0) - 1} - 1 = 2^{\mathrm{o}(h_0)} - 1 = \mathrm{o}(h)\end{aligned}$$

We now turn to the local faithfulness property of Γ which we also prove by induction on the build-up of $h \in \mathcal{CH}$. We abbreviate

$$*(h) \quad := \quad \Delta(\mathrm{tp}(h)) \cup \bigcup_{\iota < |\,\mathrm{tp}(h)|} \Big(\Gamma(h[\iota]) \setminus \approx\!\Delta_\iota(\mathrm{tp}(h))\Big) \ ,$$

then we have to show $*(h) \subseteq \approx\!\Gamma(h)$.

- If $h \in \mathcal{H}$ then the local faithfulness property is inherited from $\mathcal{H}$.
- If $h = \mathsf{I}^k_C h_0$, then $\Gamma(h) := \{C[k]\} \cup (\Gamma(h_0) \setminus \approx\!\{C\})$.

  Case 1. $\mathrm{tp}(h_0) \in \{\bigwedge_D \colon D \approx C\}$. Then $\Gamma(h_0[k]) \subseteq *(h_0) \cup \approx\!\{C[k]\}$ hence

$$\begin{aligned}*(h) &= \emptyset \cup \Gamma(\mathsf{I}^k_C h_0[k]) \\ &= \{C[k]\} \cup \Big(\Gamma(h_0[k]) \setminus \approx\!\{C\}\Big) \\ &\subseteq \{C[k]\} \cup \Big(*(h_0) \setminus \approx\!\{C\}\Big) \\ &\overset{i.h.}{\subseteq} \{C[k]\} \cup \Big(\approx\!\Gamma(h_0) \setminus \approx\!\{C\}\Big) \quad \subseteq \quad \approx\!\Gamma(h)\end{aligned}$$

Case 2. Otherwise, we compute

$$
\begin{aligned}
*(h) &= \Delta(\mathrm{tp}(h_0)) \cup \bigcup_{\iota<|\mathrm{tp}(h_0)|} \Big(\Gamma(\mathsf{I}_C^k h_0[\iota]) \setminus {\approx}\Delta_\iota(\mathrm{tp}(h_0))\Big) \\
&= \Delta(\mathrm{tp}(h_0)) \cup \bigcup_{\iota<|\mathrm{tp}(h_0)|} \Big(\Big[\{C[k]\} \cup \big(\Gamma(h_0[\iota]) \setminus {\approx}\{C\}\big)\Big] \setminus {\approx}\Delta_\iota(\mathrm{tp}(h_0))\Big) \\
&\subseteq \{C[k]\} \cup \Big(\Big[\Delta(\mathrm{tp}(h_0)) \cup \bigcup_{\iota<|\mathrm{tp}(h_0)|} \big(\Gamma(h_0[\iota]) \setminus {\approx}\Delta_\iota(\mathrm{tp}(h_0))\big)\Big] \setminus {\approx}\{C\}\Big) \\
&= \{C[k]\} \cup \Big(*(h_0) \setminus {\approx}\{C\}\Big) \\
&\overset{i.h.}{\subseteq} \{C[k]\} \cup \Big({\approx}\Gamma(h_0) \setminus {\approx}\{C\}\Big) \quad \subseteq \quad {\approx}\Gamma(h)
\end{aligned}
$$

- $h = \mathsf{R}_C h_0 h_1$: Let $\mathcal{I} := \mathrm{tp}(h_1)$. We have $\Gamma(h) := (\Gamma(h_0) \setminus {\approx}\{C\}) \cup (\Gamma(h_1) \setminus {\approx}\{\neg C\})$.

  Case 1. $\Delta(\mathcal{I}) \cap {\approx}\{\neg C\} = \emptyset$: We compute

$$
\begin{aligned}
*(h) &= \Delta(\mathcal{I}) \cup \bigcup_{\iota<|\mathcal{I}|} \Big(\Gamma(\mathsf{R}_C h_0 h_1[\iota]) \setminus {\approx}\Delta_\iota(\mathcal{I})\Big) \\
&= \Delta(\mathcal{I}) \cup \bigcup_{\iota<|\mathcal{I}|} \Big(\Big[\Gamma(h_0) \setminus {\approx}\{C\} \cup \Gamma(h_1[\iota]) \setminus {\approx}\{\neg C\}\Big] \setminus {\approx}\Delta_\iota(\mathcal{I})\Big) \\
&\subseteq \Gamma(h_0) \setminus {\approx}\{C\} \cup \Big(\Big[\Delta(\mathcal{I}) \cup \bigcup_{\iota<|\mathcal{I}|} \big(\Gamma(h_1[\iota]) \setminus {\approx}\Delta_\iota(\mathcal{I})\big)\Big] \setminus {\approx}\{\neg C\}\Big) \\
&= \Gamma(h_0) \setminus {\approx}\{C\} \ \cup \ *(h_1) \setminus {\approx}\{\neg C\} \\
&\overset{i.h.}{\subseteq} \Gamma(h_0) \setminus {\approx}\{C\} \ \cup \ {\approx}\Gamma(h_1) \setminus {\approx}\{\neg C\} \quad \subseteq \quad {\approx}\Gamma(h)
\end{aligned}
$$

  Case 2. Otherwise, we compute

$$
\begin{aligned}
*(h) &= \Gamma(\mathsf{I}_C^k h_0) \setminus {\approx}\{C[k]\} \ \cup \ \Gamma(\mathsf{R}_C h_0 h_1[0]) \setminus {\approx}\{\neg C[k]\} \\
&= \Big(\{C[k]\} \cup \big(\Gamma(h_0) \setminus {\approx}\{C\}\Big) \setminus {\approx}\{C[k]\} \\
&\quad \cup \Big(\Gamma(h_0) \setminus {\approx}\{C\} \cup \Gamma(h_1[0]) \setminus {\approx}\{\neg C\}\Big) \setminus {\approx}\{\neg C[k]\} \\
&\subseteq \Gamma(h_0) \setminus {\approx}\{C\} \ \cup \ \Big(\Gamma(h_1[0]) \setminus {\approx}\{\neg C[k]\}\Big) \setminus {\approx}\{\neg C\} \\
&\subseteq \Gamma(h_0) \setminus {\approx}\{C\} \ \cup \ *(h_1) \setminus {\approx}\{\neg C\} \\
&\overset{i.h.}{\subseteq} \Gamma(h_0) \setminus {\approx}\{C\} \ \cup \ {\approx}\Gamma(h_1) \setminus {\approx}\{\neg C\} \quad \subseteq \quad {\approx}\Gamma(h)
\end{aligned}
$$

- $h = \mathsf{E} h_0$: Then $\Gamma(h) := \Gamma(h_0)$.

Case 1. $\mathrm{tp}(h_0) = \mathrm{Cut}_C$: Assume $\mathrm{tp}(C) \in \{\top, \bigwedge\}$, then

$$\begin{aligned}
*(h) &= \Gamma(\mathsf{R}_C \mathsf{E} h_0 \mathsf{E} h_1) \\
&= \Gamma(\mathsf{E} h_0[0]) \setminus \approx\{C\} \ \cup\ \Gamma(\mathsf{E} h_0[1]) \setminus \approx\{\neg C\} \\
&= \Gamma(h_0[0]) \setminus \approx\{C\} \ \cup\ \Gamma(h_0[1]) \setminus \approx\{\neg C\} \\
&= *(h_0) \quad \overset{i.h.}{\subseteq} \quad \approx\Gamma(h_0) \quad \subseteq \quad \approx\Gamma(h)
\end{aligned}$$

The case that $\mathrm{tp}(C) \notin \{\top, \bigwedge\}$ runs similar.

Case 2. Otherwise, we compute

$$\begin{aligned}
*(h) &= \Delta(\mathrm{tp}(h_0)) \cup \bigcup_{\iota < |\,\mathrm{tp}(h_0)|} \Big(\Gamma(\mathsf{E} h_0[\iota]) \setminus \approx\Delta_\iota(\mathrm{tp}(h_0))\Big) \\
&= \Delta(\mathrm{tp}(h_0)) \cup \bigcup_{\iota < |\,\mathrm{tp}(h_0)|} \Big(\Gamma(h_0[\iota]) \setminus \approx\Delta_\iota(\mathrm{tp}(h_0))\Big) \\
&= *(h_0) \quad \overset{i.h.}{\subseteq} \quad \approx\Gamma(h_0) \quad = \quad \approx\Gamma(h)
\end{aligned}$$

∎

Remark 7.6. For the computation of Γ, the cut-elimination operators I^k_C, R_C and E behave like the following inference symbols:

$$(\mathsf{I}^k_C)\quad \frac{C}{C[k]}\ , \qquad (\mathsf{R}_C)\quad \frac{C \qquad \neg C}{\emptyset}\ , \qquad (\mathsf{E})\quad \frac{\emptyset}{\emptyset}\ .$$

Definition 7.7. Let $\mathcal{CH}$ be the notation system for cut-elimination on $\mathcal{H}$. The *interpretation* $[\![h]\!]$ is extended inductively from $\mathcal{H}$ to $\mathcal{CH}$ by defining

$$\begin{aligned}
[\![\mathsf{I}^k_C h]\!] &= \mathbb{I}^k_C([\![h]\!]) \\
[\![\mathsf{R}_C h_0 h_1]\!] &= \mathbb{R}_C([\![h_0]\!], [\![h_1]\!]) \\
[\![\mathsf{E} h]\!] &= \mathbb{E}([\![h]\!]).
\end{aligned}$$

Proposition 7.8. *For $h \in \mathcal{CH}$ we have*

$$\begin{aligned}
\mathrm{last}([\![h]\!]) &= \mathrm{tp}(h) \\
[\![h]\!](\iota) &= [\![h[\iota]]\!] \qquad \textit{for } \iota < |\,\mathrm{tp}(h)| \\
\mathcal{C}\text{-crk}([\![h]\!]) &\le \mathcal{C}\text{-crk}(h)
\end{aligned}$$

Proof. By induction on the build-up of $h \in \mathcal{CH}$. If $h \in \mathcal{H}$ then the assertion is inherited from $\mathcal{H}$ and Observation 7.4. The remaining cases follow from Theorems 6.1, 6.2 and 6.3. ∎

8 An Abstract Notion of Notation

We are now interested in studying the size needed by the notations for subderivations of derivations obtained by the cut-elimination operator. To avoid losing the simple idea in a blurb of notation, we abstract our problem to a simple term-rewriting system.

Definition 8.1. An *abstract system of proof notations* is a set $\mathcal{D}$ of "derivations", together with two functions $|\cdot|, o(\cdot)\colon \mathcal{D} \to \mathbb{N} \setminus \{0\}$, called "size" and "height", and a relation $\to \subseteq \mathcal{D} \times \mathcal{D}$ called "reduction to a sub-derivation", such that $d \to d'$ implies $o(d') < o(d)$.

Observation 8.2 (and Definition)**.** *Let $\mathcal{F}$ be a notation system for formulae and $\mathfrak{S}_{\mathcal{F}}$ the semiformal proof system over $\mathcal{F}$. A notation system $\mathcal{H} = (\mathcal{H}, \mathrm{tp}, \cdot[\cdot], \mathrm{o}, |\cdot|)$ for $\mathfrak{S}_{\mathcal{F}}$ gives rise to an abstract system of proof notations by letting $\mathcal{D} = \mathcal{H}$ and defining $d \to d'$ iff there exists an $n < |\operatorname{tp}(d)|$ with $d' = d[n]$.*

Definition 8.3. If $\mathcal{D}$ is an abstract system of proof notations, then $\widetilde{\mathcal{D}}$, the "cut elimination closure", is the abstract notation system extending $\mathcal{D}$ that is inductively defined by

$$\frac{d \in \mathcal{D}}{d \in \widetilde{\mathcal{D}}} \qquad \frac{d \in \widetilde{\mathcal{D}}}{\mathsf{I}d \in \widetilde{\mathcal{D}}} \qquad \frac{d \in \widetilde{\mathcal{D}} \qquad e \in \widetilde{\mathcal{D}}}{\mathsf{R}de \in \widetilde{\mathcal{D}}} \qquad \frac{d \in \widetilde{\mathcal{D}}}{\mathsf{E}d \in \widetilde{\mathcal{D}}}$$

$$|\mathsf{I}d| = |d| + 1 \qquad |\mathsf{R}de| = |d| + |e| + 1 \qquad |\mathsf{E}d| = |d| + 1$$

$$\frac{d \to d' \text{ in } \mathcal{D}}{d \to d'} \qquad \frac{d \to d'}{\mathsf{I}d \to \mathsf{I}d'} \qquad \frac{e \to e'}{\mathsf{R}de \to \mathsf{R}de'} \qquad \frac{d \to d'}{\mathsf{E}d \to \mathsf{E}d'}$$

$$\frac{}{\mathsf{R}de \to \mathsf{I}d} \qquad \frac{d \to d' \qquad d \to d''}{\mathsf{E}d \to \mathsf{R}(\mathsf{E}d')(\mathsf{E}d'')}$$

$$o(\mathsf{I}d) = o(d) \qquad o(\mathsf{R}de) = o(d) + o(e) \qquad o(Ed) = 2^{o(d)} - 1$$

where E, R, I are new symbols.

Proof. We have to show that whenever $d \to d'$ for $d, d' \in \widetilde{\mathcal{D}}$ then $o(d) > o(d')$. We show this by Induction following the inductive definition of the $\to$ relation in $\widetilde{\mathcal{D}}$. If $d \to d'$ holds in $\widetilde{\mathcal{D}}$ because it already holds in $\mathcal{D}$ then $o(d) > o(d')$ is inherited from $\mathcal{D}$. The cases $\mathsf{I}d \to \mathsf{I}d'$, $\mathsf{R}de \to \mathsf{R}de'$ and $\mathsf{E}d \to \mathsf{E}d'$ are immediate by induction hypothesis.

For the remaining cases we argue as follows. In case $\mathsf{R}de \to \mathsf{I}d$ we calculate $o(\mathsf{R}de) = o(d) + o(e) > o(d) = o(\mathsf{I}d)$, since $o(e) > 0$.

In the case $\mathsf{E}d \to \mathsf{R}(Ed')(Ed'')$ thanks to $d \to d'$ and $d \to d''$ we have $o(d) \geq o(d') + 1$ and $o(d) \geq o(d'') + 1$. So, we calculate $o(\mathsf{R}(\mathsf{E}d')(\mathsf{E}d'')) = o(\mathsf{E}d') + o(\mathsf{E}d'') = 2^{o(d')} - 1 + 2^{o(d'')} - 1 < 2^{o(d')} + 2^{o(d'')} - 1 \leq 2^{o(d)} - 1 = o(\mathsf{E}d)$. ∎

Let $\mathcal{F}$ be a notation system for formulae, $\mathfrak{S}_{\mathcal{F}}$ the semiformal proof system over $\mathcal{F}$, $\mathcal{H}$ a notation system for $\mathfrak{S}_{\mathcal{F}}$, $\mathcal{CH}$ the notation system for cut-elimination

on $\mathcal{H}$ with denoted height o and size $|\cdot|$, and let $\mathcal{D}$ be the abstract system of proof notations associated with $\mathcal{H}$ according to Observation 8.2.

Definition 8.4. The abstraction $\overline{h}$ of $h \in \mathcal{CH}$ is obtained by dropping all sub- and superscripts. It can be defined by induction on the build-up of $h \in \mathcal{CH}$:

- $h \in \mathcal{H} \quad \Rightarrow \quad \overline{h} := h$,
- $h = \mathsf{I}^k_C h_0 \quad \Rightarrow \quad \overline{h} := \mathsf{I}\overline{h_0}$,
- $h = \mathsf{R}_C h_0 h_1 \quad \Rightarrow \quad \overline{h} := \mathsf{R}\,\overline{h_0}\,\overline{h_1}$,
- $h = \mathsf{E}h_0 \quad \Rightarrow \quad \overline{h} := \mathsf{E}\overline{h_0}$.

We denote the set of abstractions for $h \in \mathcal{CH}$ by $\overline{\mathcal{CH}}$.

Observation 8.5 (and Definition)**.** *The set of abstractions $\overline{\mathcal{CH}}$ for $\mathcal{CH}$ is a subsystem of the cut-elimination closure $\widetilde{\mathcal{H}}$ of $\mathcal{H}$ in the following sense: Let $\rightarrow$ denote the reduction to sub-derivation relation of $\widetilde{\mathcal{H}}$, and define a reduction to sub-derivation relation $\rightsquigarrow$ of $\overline{\mathcal{CH}}$ in the obvious way by $\overline{h} \rightsquigarrow \overline{h'}$ iff there exists an $n < |\operatorname{tp}(h)|$ with $h' = h[n]$. Then $\overline{\mathcal{CH}} = \widetilde{\mathcal{H}}$ and $\rightsquigarrow \subseteq \rightarrow$.*

9 Size Bounds

We now prove a bound on the size of (abstract) notations for cut-elimination. By induction on the build up of $\widetilde{\mathcal{D}}$ we assign every element a measure that bounds the size of all derivations reachable from it via iterated use of the $\rightarrow$-relation. A small problem arises in the base case; if $d \rightarrow d'$ in $\widetilde{\mathcal{D}}$ because this holds in $\mathcal{D}$ we have no means of bounding $|d'|$ in terms of $|d|$. So we use the usual trick [AS00] when a global measure is needed and assign each element d of $\widetilde{\mathcal{D}}$ not a natural number but a monotone function $\vartheta(d)$ such that $|d'| \leq \vartheta(d)(s)$ for all $d \rightarrow^* d'$ whenever $s \in \mathbb{N}$ is a global bound on the size of all elements in $\mathcal{D}$.

Definition 9.1. An abstract system $\mathcal{D}$ of proof notations is called s-bounded (for $s \in \mathbb{N}$), if for all $d \in \mathcal{D}$ it is the case that $|d| \leq s$.

Definition 9.2. If $\mathcal{D}$ is an abstract system of proof notations and $d \in \mathcal{D}$, then by $\mathcal{D}_d$ we denote the set $\mathcal{D}_d = \{d' \mid d \rightarrow^* d'\} \subset \mathcal{D}$ considered an abstract system of proof notation with the structure induced by $\mathcal{D}$. Here $\rightarrow^*$ denotes the reflexive transitive closure of $\rightarrow$.

Definition 9.3. For $\mathcal{D}$ an abstract system of proof notations and $d \in \mathcal{D}$ we say that d is s-bounded if $\mathcal{D}_d$ is.

Definition 9.4. By $\mathfrak{S}$ we denote the set of all monotone functions from $\mathbb{N}$ to $\mathbb{N}$.

Definition 9.5. For $\mathcal{D}$ an abstract system of proof notations we define, a "size function" $\vartheta(d) \in \mathfrak{S}$ for every $d \in \widetilde{\mathcal{D}}$ by induction on the inductive definition of $\widetilde{\mathcal{D}}$ as follows.

- For $d \in \mathcal{D}$ we set $\vartheta(d)(s) = s$.
- $\vartheta(\mathsf{I}d)(s) = \vartheta(d)(s) + 1$
- $\vartheta(\mathsf{R}de)(s) = \max\{|d|+1+\vartheta(e)(s)\ ,\ \vartheta(d)(s)+1\}$
- $\vartheta(\mathsf{E}d)(s) = o(d)(\vartheta(d)(s) + 2)$

Proof. The monotonicity of the defined function $\vartheta(d)$ is immediately seen from the definition and the induction hypothesis. ◻

Proposition 9.6. *If $\mathcal{D}$ is s-bounded then for every $d \in \widetilde{\mathcal{D}}$ we have $|d| \leq \vartheta(d)(s)$.*

Proof. By induction on the inductive definition of $\widetilde{\mathcal{D}}$.

If $d \in \mathcal{D}$ then $\vartheta(d)(s) = s \geq |d|$, since $\mathcal{D}$ is s-bounded. We calculate $\vartheta(\mathsf{I}d)(s) = \vartheta(d)(s) + 1 \geq |d| + 1 = |\mathsf{I}d|$, where we used that $\vartheta(d)(s) \geq |d|$ by induction hypothesis. Also, $\vartheta(\mathsf{R}de)(s) \geq |d|+1+\vartheta(e)(s) \geq 1+|d|+|e| = |\mathsf{R}de|$, using the induction hypothesis for e. Finally, $\vartheta(\mathsf{E}d)(s) = o(d)(\vartheta(d)(s) + 2) \geq \vartheta(d)(s)+1 \geq |d|+1 = |\mathsf{E}d|$, where for the first inequality we used that $o(d) \geq 1$, and for the second inequality we used the induction hypothesis. ◻

Theorem 9.7. *If $\mathcal{D}$ is s-bounded, $d \in \widetilde{\mathcal{D}}$ and $d \to d'$, then $\vartheta(d)(s) \geq \vartheta(d')(s)$.*

Proof. Induction on the inductive definition of the relation $d \to d'$ in $\widetilde{\mathcal{D}}$.

If $d \to d'$ because it holds in $\mathcal{D}$ then $\vartheta(d)(s) = s = \vartheta(d')(s)$.

If $\mathsf{I}d \to \mathsf{I}d'$ thanks to $d \to d'$ then $\vartheta(\mathsf{I}d)(s) = \vartheta(d)(s) + 1 \geq \vartheta(d')(s) + 1 = \vartheta(\mathsf{I}d')(s)$, where the inequality is due to the induction hypothesis.

If $\mathsf{E}d \to \mathsf{R}(\mathsf{E}d')(\mathsf{E}d'')$ thanks to $d \to d'$ and $d \to d''$ we argue as follows

$$
\begin{array}{ll}
\multicolumn{2}{l}{\vartheta(\mathsf{R}(\mathsf{E}d')(\mathsf{E}d''))(s)} \\
= & \max\{|\mathsf{E}d'|+1+\vartheta(\mathsf{E}d'')(s)\ ,\ \vartheta(\mathsf{E}d')(s)+1\} \\
= & \max\{|d'|+2+o(d'')(\vartheta(d'')(s)+2)\ ,\ o(d')(\vartheta(d')(s) + 2)\} \\
\leq & \max\{\vartheta(d')(s)+2+o(d'')(\vartheta(d'')(s)+2)\ ,\ o(d')(\vartheta(d')(s) + 2)\} \\
\leq & \max\{\vartheta(d)(s)+2+o(d'')(\vartheta(d)(s)+2)\ ,\ o(d')(\vartheta(d)(s) + 2)\} \\
\leq & \max\{\vartheta(d)(s)+2+(o(d)-1)(\vartheta(d)(s)+2)\ ,\ (o(d)-1)(\vartheta(d)(s) + 2)\} \\
= & \vartheta(d)(s)+2+(o(d)-1)(\vartheta(d)(s)+2) \\
= & o(d)(\vartheta(d)(s)+2) \\
= & \vartheta(\mathsf{E}d)(s)
\end{array}
$$

where for the first inequality we used Proposition 9.6, for the second the induction hypothesis, for the third that, since $d \to d'$ and $d \to d''$, both $o(d')$ and $o(d'')$ are bounded by $o(d) - 1$.

If $\mathsf{E}d \to \mathsf{E}d'$ thanks to $d \to d'$ then $\vartheta(\mathsf{E}d')(s) = o(d')(\vartheta(d')(s) + 2) \leq o(d)(\vartheta(d')(s) + 2) \leq o(d)(\vartheta(d)(s) + 2) = \vartheta(\mathsf{E}d)(s)$.

If $\mathsf{R}de \to \mathsf{R}de'$ thanks to $e \to e'$, then

$$
\begin{array}{ll}
& \vartheta(\mathsf{R}de')(s) \\
= & \max\{|d|+1+\vartheta(e')(s)\ ,\ \vartheta(d)(s)+1\} \\
\leq & \max\{|d|+1+\vartheta(e)(s)\ ,\ \vartheta(d)(s)+1\} \\
= & \vartheta(\mathsf{R}de)
\end{array}
$$

where for the inequality we used the induction hypothesis.

If $\mathsf{R}de \to \mathsf{I}d$ then $\vartheta(\mathsf{R}de)(s) \geq \vartheta(d)(s) + 1 = \vartheta(\mathsf{I}d)(s)$. ∎

Now we draw the desired consequences of our main theorem by putting things together.

Lemma 9.8. *If $\mathcal{D}$ is s-bounded, and $d \in \widetilde{\mathcal{D}}$ then $\widetilde{\mathcal{D}}_d$ is $\vartheta(d)(s)$-bounded.*

Proof. We first show by induction on the inductive definition of the reflexive transitive closure that for every $d' \in \widetilde{\mathcal{D}}_d = \{d' \in \widetilde{\mathcal{D}} \mid d \to^* d'\}$ we have $\vartheta(d)(s) \geq \vartheta(d')(s)$. The case $d = d'$ is trivial and if $d \to^* d' \to d''$ then $\vartheta(d)(s) \geq \vartheta(d')(s)$ by induction hypothesis and $\vartheta(d')(s) \geq \vartheta(d'')(s)$ by Theorem 9.7.

Now, by Proposition 9.6 we know that $\vartheta(d')(s) \geq |d'|$ for $d' \in \widetilde{\mathcal{D}}$. So, with the previous claim, for $d' \in \widetilde{\mathcal{D}}_d$ we get $\vartheta(d)(s) \geq \vartheta(d')(s) \geq |d'|$, which is the claim. ∎

Corollary 9.9. *If $d \in \mathcal{D}$ is s-bounded then $\mathsf{E}d$ is $o(d)(s+2)$-bounded and $\mathsf{EE}d$ is $2^{o(d)} \cdot o(d) \cdot (s+4)$-bounded.*

Proof. Let $d \in \mathcal{D}$ be s-bounded and $h := o(d)$. First we observe that $\widetilde{(\mathcal{D}_d)}_{d'} = \widetilde{\mathcal{D}}_{d'}$ for any $d' \in \widetilde{(\mathcal{D}_d)}$. So we can assume without loss of generality that $\mathcal{D}$ is s, h-bounded.

Lemma 9.8 now gives us that $\mathsf{E}d$ is $\vartheta(\mathsf{E}d)(s)$-bounded and $\mathsf{EE}d$ is $\vartheta(\mathsf{EE}d)(s)$-bounded. We calculate $\vartheta(\mathsf{E}d) = o(d)(\vartheta(d)(s)+2) = o(d)(s+2) \leq h(s+2)$ and $\vartheta(\mathsf{EE}d) = o(\mathsf{E}d)(\vartheta(\mathsf{E}d)(d)+2) = o(Ed)(h(s+2)+2) \leq (2^h - 1)(h(s+2)+2) \leq 2^h \cdot h \cdot (s+4)$. ∎

Even though the above Corollary covers all the case usually needed in practise, it is interesting to consider the general case. Recall that iterated exponentiation $2_n(x)$ is defined inductively by setting $2_0(x) = x$ and $2_{n+1}(x) = 2^{2_n(x)}$. An easy induction shows that the height $o(E^n d)$ of the n-times cut-reduced derivation d is bounded by $2_n(d)$.

Lemma 9.10. *$\vartheta(E^n d)(s) \leq 2_{n-1}(2 \cdot o(d)) \cdot s$ for all $n \geq 1$, $s \geq 2$ and $o(d) \geq 2$.*

Proof. Induction on n. For the case $n = 1$ we compute $\vartheta(Ed)(s) = o(d)(s+2) \leq 2o(d)s$.

For $n = 2$ we compute $\vartheta(EEd)(s) = (2^{o(d)} - 1)(o(d)(s+2)+2)$. For $o(d) = 2$ and $o(d) = 3$ we directly compute that this is bounded by $2^{2o(d)}s$. For $o(d) \geq 4$ we compute $\vartheta(EEd)(s) \leq 2^{o(d)} 4o(d)s \leq 2^{2o(d)}s$.

Now assume that the claim holds for $n \geq 2$. We then compute $\vartheta(EE^n d)(s) = o(E^n d)(\vartheta(E^n d)(s)+2) \leq 2_{n-1}(2^{o(d)} - 1) \cdot (2_{n-1}(2 \cdot o(d)) \cdot s + 2) \leq 2_{n-1}(2^{o(d)} - 1) \cdot 2 \cdot 2_n(o(d)) \cdot s \leq 2_n(o(d)) \cdot 2_n(o(d)) \cdot s \leq 2_n(2 \cdot o(d)) \cdot s$ ∎

As an immediate Corollary we obtain

Corollary 9.11. *If $d \in \mathcal{D}$ is s-bounded of height $o(d) = h$ for $s \geq 2$ and $h \geq 2$, then $E^k(d)$ is $2_{k-1}(2 \cdot h) \cdot s$-bounded for all $k \geq 1$.*

In Corollary 9.11 one should note that the tower of exponentiations has height only $k-1$. Hence there is one exponentiation less than the height of the denoted proof.

We conclude this section by remarking that the cut-elimination operator can be viewed as a polynomial time computable operation. Assume we modify the size function on $\widetilde{\mathcal{D}}$ to ϑ_k by changing all ϑ to ϑ_k and defining for the last case

- $\vartheta_k(\mathsf{E}d)(s) = (k+1) \cdot (\vartheta(d)(s) + 2)$

Then we obtain as before for $\mathcal{D}$ s-bounded, $d \in \widetilde{\mathcal{D}}$ and $k \in \mathbb{N}$, that $|d| \leq \vartheta_k(d)(s)$, and $d \to d'$ implies $\vartheta_{k+1}(d)(s) \geq \vartheta_k(d)(s)$. Hence, for $d \in \mathcal{D}$, $\mathcal{D}$ s-bounded, and $\mathsf{E}d \to^k d'$, we obtain $|d'| \leq \vartheta_k(\mathsf{E}d)(s) \leq (k+1) \cdot (s+2)$. From this we can conclude the following observation: Let $f[i_1, \ldots, i_k] := f[i_1] \ldots [i_k]$.

Observation 9.12. *The cut-reduction operator for infinitary propositional logic is a polynomial time operation in the following sense.*

Let $\mathcal{F}$ and $\mathcal{H}$ be some notation systems for infinitary formulae and the semi-formal system $\mathfrak{S}_{\mathcal{F}}$. Assume that $\mathcal{F}$ and $\mathcal{H}$ are polynomial time computable, and that in addition also the functions

$$\begin{aligned} \mathcal{F} \times \mathbb{N}^{<\omega} &\to \mathcal{F} \\ A, (i_1, \ldots, i_k) &\mapsto A[i_1, \ldots, i_k] \end{aligned}$$

and

$$\begin{aligned} \mathcal{H} \times \mathbb{N}^{<\omega} &\to \mathcal{H} \\ h, (i_1, \ldots, i_k) &\mapsto h[i_1, \ldots, i_k] \end{aligned}$$

are polynomial time computable.

Then, $\mathcal{CH}$ and the function

$$\begin{aligned} \mathcal{H} \times \mathbb{N}^{<\omega} &\to \mathcal{CH} \\ h, (i_1, \ldots, i_k) &\mapsto (\mathsf{E}h)[i_1, \ldots, i_k] \end{aligned}$$

are polynomial time computable.

10 Bounded Arithmetic

Our proof-theoretic investigations are very much independent of the exact choice of the language. Therefore, we will be very liberal and allow symbols for all ptime functions.

Definition 10.1 (Language of Bounded Arithmetic)**.** *The language* $\mathcal{L}_{\mathrm{BA}}$ *of Bounded Arithmetic* contains as non-logical symbols $\{=, \leq\}$ for the binary relation "equality" and "less than or equal", and a symbol for each ptime function. In particular, it includes a constant c_a for $a \in \mathbb{N}$ whose interpretation in the standard model $\mathbb{N}$ is $c_a^{\mathbb{N}} = a$, unary function symbols $|\cdot|$ and $2^{|\cdot|}$ which have their standard interpretation given by $(|c_a|)^{\mathbb{N}} = n$ and $(2^{|c_a|})^{\mathbb{N}} = 2^n$ where n is the length of the binary representation of a, and the binary function symbols min and $\#$ whose standard interpretation are minimisation and $(c_a \# c_b)^{\mathbb{N}} = 2^{n \cdot m}$ where n and m are the lengths of the binary representations of a resp. b. We will often write $\underline{n}$ instead of c_n, and 0 for c_0.

Atomic formulae are of the form $s = t$ or $s \leq t$ where s and t are terms. *Literals* are expressions of the form A or $\neg A$ where A is an atomic formula. Formulas are build up from literals by means of $\wedge$, $\vee$, $(\forall x)$, $(\exists x)$. The *negation* $\neg C$ *for a formula* C is defined via de Morgan's laws. Negation extends to sets of formulae in the usual way by applying it to their members individually.

Let $\mathcal{C}$ be a set of $\mathcal{L}_{\mathrm{BA}}$-formulae (think of Σ_i^{b}), and A an $\mathcal{L}_{\mathrm{BA}}$-formula. We define the *$\mathcal{C}$-rank of A,* denoted $\mathcal{C}\text{-rk}(A)$, by induction on the build-up of A:

- If $A \in \mathcal{C} \cup \neg\mathcal{C}$, let $\mathcal{C}\text{-rk}(A) := 0$.
- If $A = B \wedge C$ or $A = B \vee C$, let $\mathcal{C}\text{-rk}(A) := 1 + \max\{\mathcal{C}\text{-rk}(B), \mathcal{C}\text{-rk}(C)\}$.
- If $A = (\forall x)B$ or $A = (\exists x)B$, let $\mathcal{C}\text{-rk}(A) := 1 + \mathcal{C}\text{-rk}(B)$.

We will use the following standard abbreviations.

Definition 10.2 (Abbreviations)**.** The expression $A \rightarrow B$ denotes the expression $\neg A \vee B$. The expression $s < t$ denotes $\neg t \leq s$. Bounded quantifiers are introduced as follows: $(\forall x \leq t)A$ denotes $(\forall x)A_x(\min(x,t))$, $(\exists x \leq t)A$ denotes $(\exists x)A_x(\min(x,t))$, $(\forall x < t)A$ denotes $(\forall x \leq t)(x < t \rightarrow A)$, $(\exists x < t)A$ denotes $(\exists x \leq t)(x < t \wedge A)$, where x may not occur in t.

Definition 10.3 (Bounded Formulas)**.** The set BFOR of bounded $\mathcal{L}_{\mathrm{BA}}$-formulae is the set of $\mathcal{L}_{\mathrm{BA}}$-formulae consisting of literals and closed under $\wedge$, $\vee$, $(\forall x \leq t)$, $(\exists x \leq t)$.

We now define a restricted (also called "strict") delineation of bounded formulae.

Definition 10.4. The set $\mathrm{s}\Sigma_d^{\mathrm{b}}$ is the subset of bounded $\mathcal{L}_{\mathrm{BA}}$-formulae whose elements are of the form

$$(\exists x_1 \leq t_1)(\forall x_2 \leq t_2) \ldots (Q x_d \leq t_d)(\bar{Q} x_{d+1} \leq |t_{d+1}|)A(\vec{x})$$

with Q and $\bar{Q}$ being of the corresponding alternating quantifier shape, and A being quantifier free.

Definition 10.5. As axioms we allow all disjunctions of literals, i.e., all disjunctions A of literals such that A is true in $\mathbb{N}$ under any assignment. Let us denote this set of axioms by BASIC.

We will base the definition of Bounded Arithmetic theories on a somewhat stronger normal form of induction. Let $|\cdot|_m$ denote the m-fold iteration of the function symbol $|\cdot|$.

Definition 10.6. Let $\mathrm{Ind}(A, z, t)$ denote the expression

$$A_z(0) \ \wedge\ (\forall z < t)(A \ \rightarrow\ A_z(\mathrm{s}\, z)) \ \rightarrow\ A_z(t) \ .$$

The set Φ-$\mathrm{L}^m\mathrm{IND}$ consists of all expressions of the form

$$\mathrm{Ind}(A, z, 2^{||t|_m|})$$

with $A \in \Phi$, z a variable and t an $\mathcal{L}_{\mathrm{BA}}$-term.

This restricted form of induction implies the usual form, because the following can be proven from BASIC alone.

$$\mathrm{Ind}(A(\min(t, z)), z, 2^{|t|}) \ \rightarrow\ \mathrm{Ind}(A(z), z, t)$$

11 Notation system for Bounded Arithmetic formulae

Let $\mathcal{F}_{\mathrm{BA}}$ be the set of closed formulae in BFOR. We define the outermost connective function on $\mathcal{F}_{\mathrm{BA}}$ by

$$\mathrm{tp}(A) := \begin{cases} \top & A \text{ true literal} \\ \bot & A \text{ false literal} \\ \bigwedge & A \text{ is of the form } A_0 \ \wedge\ A_1 \text{ or } (\forall x)B \\ \bigvee & A \text{ is of the form } A_0 \ \vee\ A_1 \text{ or } (\exists x)B \ , \end{cases}$$

and the sub-formula function on $\mathcal{F}_{\mathrm{BA}} \times \mathbb{N}$ by

$$A[n] := \begin{cases} A & A \text{ literal} \\ A_{\min(n,1)} & A \text{ is of the form } A_0 \ \wedge\ A_1 \text{ or } A_0 \ \vee\ A_1 \\ B_x(\underline{n}) & A \text{ is of the form } (\forall x)B \text{ or } (\exists x)B \ . \end{cases}$$

The rank and negation functions for the notation system are those defined for $\mathcal{L}_{\mathrm{BA}}$.

We didn't have much choice on how to render BFOR into a notation system for formulae. Nevertheless, the above definition already shows that we have to work with a non-trivial intensional equality. The reason is that, even though in the process of the propositional translation we can make sure that we only have closed formulae, this still is not enough; we do have other closed terms than just the canonical ones.

Consider, for example, an arithmetical derivation ending in

$$\frac{\begin{array}{c}\vdots\\ B(f(\underline{0}))\end{array}}{\exists x.B(x)}$$

where f is some function symbol. In the propositional translation we have to provide some witness i for the $\bigvee^i_{\exists x.B(x)}$-inference. The "obvious" choice seems to take $i = f^{\mathbb{N}}(0)$. But this would require a derivation of $(\exists x.B(x))[f^{\mathbb{N}}(0)] = B(\underline{f^{\mathbb{N}}(0)})$. The translation of the sub-derivation, on the other hand, gives us a derivation of $B(f(\underline{0}))$. So, in order to make this a correct inference in the propositional translation, he have to consider $B(f(\underline{0}))$ and $B(\underline{f^{\mathbb{N}}(0)})$ as intensionally equal. Note that both formulae are extensionally equal.

We will now define an intensional equality which provides the above described identification. For t a closed term its numerical value $t^{\mathbb{N}} \in \mathbb{N}$ is defined in the obvious way. Let $\to^1_{\mathbb{N}}$ denote the rewriting relation obtained from

$$\left\{(t, \underline{t^{\mathbb{N}}})\colon t \text{ a closed term}\right\} \quad .$$

For example,

$$(\forall x)(x \leq \lfloor \tfrac{1}{2}(\underline{5} \cdot \underline{3}) \rfloor) \to^1_{\mathbb{N}} (\forall x)(x \leq \underline{7}) \quad .$$

Let $\approx_{\mathbb{N}}$ denote the reflexive, symmetric and transitive closure of $\to^1_{\mathbb{N}}$.

Proposition 11.1. *The just defined system consisting of $\mathcal{F}_{\mathrm{BA}}$, tp, $\cdot[\cdot]$, $\neg$, rk and $\approx_{\mathbb{N}}$ forms a notation system for formulae in the sense of Definition 4.1.*

Remark 11.2. It is an open problem what the complexity of $\approx_{\mathbb{N}}$ is (assuming a usual feasible arithmetisation of syntax). However, if the depth of expressions is restricted, and the number of function symbols representing polynomial time functions is also restricted to a finite subset, then the relation $\approx_{\mathbb{N}}$ is polynomial time decidable. I.e., let ${\approx_{\mathbb{N}}}^k$ denote the restriction of $\approx_{\mathbb{N}}$ to expressions of depth $\leq k$ in which at most the first k function symbols occur. Then, for each k, the relation ${\approx_{\mathbb{N}}}^k$ is a polynomial time predicate.

From now on, we will assume that $\mathcal{F}_{\mathrm{BA}}$ implicitly contains such a constant k without explicitly mentioning it. All formulae and terms used in $\mathcal{F}_{\mathrm{BA}}$ are thus assumed to obey the abovementioned restriction on occurrences of function symbols and depth. We will come back to this restriction at relevant places. The next observation already makes use of this assumption.

Observation 11.3. *All relations and functions in $\mathcal{F}_{\mathrm{BA}}$ are polynomial time computable.*

Proof. Under the just fixed convention, the relation $\approx_{\mathbb{N}}$ is actually ${\approx_{\mathbb{N}}}^k$ for some k. ∎

Definition 11.4. Let BA^∞ denote the semiformal proof system over $\mathcal{F}_{\mathrm{BA}}$ according to Definition 5.1.

12 A notation system for BA^∞

Definition 12.1. The *finitary proof system* $\mathrm{BA}^\star$ is the proof system over $\mathrm{BFOR}, \approx_{\mathbb{N}}$ which is given by the following set of inference symbols.

$$(\mathrm{Ax}_\Delta) \quad \frac{}{\Delta} \quad \text{if } \bigvee \Delta \in \mathrm{BASIC}$$

$$(\textstyle\bigwedge_{A_0 \wedge A_1}) \quad \frac{A_0 \qquad A_1}{A_0 \wedge A_1} \qquad\qquad (\textstyle\bigvee^k_{A_0 \vee A_1}) \quad \frac{A_k}{A_0 \vee A_1} \quad (k \in \{0,1\})$$

$$(\textstyle\bigwedge^y_{(\forall x)A}) \quad \frac{A_x(y)}{(\forall x)A} \qquad\qquad (\textstyle\bigvee^t_{(\exists x)A}) \quad \frac{A_x(t)}{(\exists x)A}$$

$$(\mathrm{IND}^{y,t}_F) \quad \frac{\neg F, F_y(\mathrm{s}\, y)}{\neg F_y(0), F_y(2^{|t|})} \qquad\qquad (\mathrm{IND}^{y,n,i}_F) \quad \frac{\neg F, F_y(\mathrm{s}\, y)}{\neg F_y(\underline{n}), F_y(\underline{n + 2^i})} \quad (n, i \in \mathbb{N})$$

$$(\mathrm{Cut}_C) \quad \frac{C \qquad \neg C}{\emptyset}$$

According to Definition 2.4, a $\mathrm{BA}^\star$-quasi derivation h is equipped with functions $\Gamma(h)$ denoting the endsequent of h, $\mathrm{hgt}(h)$ denoting the height of h, and $\mathrm{sz}(h)$ denoting the size of h.

In our finitary proof system Schütte's ω-rule [Sch51] is replaced by rules with Eigenvariable conditions. Of course, the precise name of the Eigenvariable does not matter, as long as it *is* an Eigenvariable. For this reason, we think of the inference symbols $\bigwedge^y_{(\forall x)A}$, $\mathrm{IND}^{y,t}_F$, and $\mathrm{IND}^{y,n,i}_F$ in $\mathrm{BA}^\star$-quasi derivations as binding the variable y in the respective sub-derivations. Fortunately, we don't have to make this intuition precise, as we will always substitute only closed (arithmetical) terms into $\mathrm{BA}^\star$-derivations and therefore no renaming of bound variables will be necessary; hence we don't have to define what this renaming would mean. Note, however, that the details of Definition 12.2 of $\mathrm{BA}^\star$-derivations and Definition 12.4 of substitution become obvious with this intuition on mind.

Definition 12.2 (Inductive definition of $\vec{x} \colon d$)**.** For $\vec{x}$ a finite list of disjoint variables and $d = \mathcal{I} d_0 \dots d_{n-1}$ a $\mathrm{BA}^\star$-quasi-derivation we inductively define the relation $\vec{x} \colon d$ that d is a $\mathrm{BA}^\star$-derivation with free variables among $\vec{x}$ as follows.

- If $\vec{x}, y \colon h_0$ and $\mathcal{I} \in \{\bigwedge^y_{(\forall x)A}, \mathrm{IND}^{y,t}_F, \mathrm{IND}^{y,n,i}_F\}$ for some A, F, t, n, i, and $\mathrm{FV}(\Gamma(\mathcal{I}h_0)) \subset \{\vec{x}\}$ then $\vec{x} \colon \mathcal{I}h_0$.
- If $\vec{x} \colon h_0$ and $\mathrm{FV}((\exists x)A), \mathrm{FV}(t) \subseteq \{\vec{x}\}$ then $\vec{x} \colon \bigvee^t_{(\exists x)A} h_0$.
- If $\vec{x} \colon h_0$, $\vec{x} \colon h_1$ and $\mathrm{FV}(C) \subseteq \{\vec{x}\}$ then $\vec{x} \colon \mathrm{Cut}_C h_0 h_1$.
- If $\mathrm{FV}(\Delta) \subseteq \{\vec{x}\}$ then $\vec{x} \colon \mathrm{Ax}_\Delta$,
- If $\vec{x} \colon h_0$, $\vec{x} \colon h_1$ and $\mathcal{I} = \bigwedge_{A_0 \wedge A_1}$ with $\mathrm{FV}(A_0 \wedge A_1) \subset \{\vec{x}\}$ then $\vec{x} \colon \mathcal{I} h_0 h_1$.
- If $\vec{x} \colon h_0$ and $\mathcal{I} = \bigvee^k_{A_0 \vee A_1}$ with $\mathrm{FV}(A_0 \vee A_1) \subset \{\vec{x}\}$ then $\vec{x} \colon \mathcal{I} h_0$.

A $\mathrm{BA}^\star$-derivation is a $\mathrm{BA}^\star$-quasi derivation h such that for some $\vec{x}$ it holds $\vec{x}\colon h$. We call a $\mathrm{BA}^\star$-derivation h *closed*, if $\emptyset\colon h$.

Proposition 12.3. *If* $\vec{x}\colon h$ *then* $\mathrm{FV}(\Gamma(h)) \subseteq \{\vec{x}\}$*. In particular* $\mathrm{FV}(\Gamma(h)) = \emptyset$ *for closed* h*.*

Proof. Trivial induction on the inductive definition of $\vec{x}\colon h$. ∎

Definition 12.4. For h a $\mathrm{BA}^\star$-derivation, y a variable and t a closed term of Bounded Arithmetic we define the substitution $h(t/y)$ inductively by setting $(\mathcal{I}h_0 \dots h_{n-1})(t/y)$ to be $\mathcal{I}(t/y)h_0(y/t)\dots h_{n-1}(t/y)$ if $\mathcal{I}$ is not of the form $\bigwedge^y_{(\forall x)A}$, $\mathrm{IND}^{y,t}_F$, or $\mathrm{IND}^{y,n,i}_F$ with the same variable y, and $\mathcal{I}h_0\dots h_{n-1}$ otherwise.

Substitution for inference symbols is defined by setting

$$
\begin{array}{lcllcl}
\mathrm{Ax}_\Delta(t/y) & = & \mathrm{Ax}_{\Delta(t/y)} & & & \\
\bigwedge_{A_0\wedge A_1}(t/y) & = & \bigwedge_{(A_0\wedge A_1)(t/y)} & \bigvee^k_{A_0\wedge A_1}(t/y) & = & \bigvee^k_{(A_0\wedge A_1)(t/y)} \\
\bigwedge^z_{(\forall x)A}(t/y) & = & \bigwedge^z_{((\forall x)A)(t/y)} & \bigvee^{t'}_{(\exists x)A}(t/y) & = & \bigvee^{t'(t/y)}_{((\exists x)A)(t/y)} \\
\mathrm{IND}^{z,t'}_F(t/y) & = & \mathrm{IND}^{z,t'(t/y)}_{F(t/y)} & \mathrm{IND}^{z,n,i}_F(t/y) & = & \mathrm{IND}^{z,n,i}_{F(t/y)}
\end{array}
$$

We now show the substitution property for $\mathrm{BA}^\star$-derivations. The formulation of Lemma 12.5 might look a bit strange with "$\subseteq$" instead of the more familiar equality. The reason is, that a substitution may make formulae equal which are not equal without the substitution.

Recalling however Definition 5.3, we note that derivations h in fact prove every superset of $\Gamma(h)$. Of course, an easy consequence of Lemma 12.5 is that if $\Gamma(h) \subset \Delta$ then $\Gamma(h(t/y)) \subset \Delta(t/y)$.

Lemma 12.5. *Assume* $\vec{x}\colon h$ *and let* y *be a variable and* t *a closed term, then* $\vec{x}\setminus\{y\}\colon h(t/y)$ *and moreover* $\Gamma(h(t/y)) \subseteq (\Gamma(h))(t/y)$*.*

Proof. We argue by induction on the build-up of h.

In the cases where no substitution occurs (as $h = \mathcal{I}\dots$ with $\mathcal{I}$ of the form $\bigwedge^y_{(\forall x)A}$, $\mathrm{IND}^{y,t}_F$, or $\mathrm{IND}^{y,n,i}_F$ with the same variable y) both claims are trivial.

Otherwise, by induction hypothesis, we know that the sub-derivations are $\mathrm{BA}^\star$-derivations with the correct set of free variables; since substitution is also carried out in the inference symbols, the y in the variable conditions for Cut_C and $\bigvee^t_{(\exists x)A}$ will also disappear due to the substitution. The Eigenvariable condition $z \notin \mathrm{FV}(\Gamma(h))$ will follow once we have shown the second claim.

For the second claim we compute by induction hypothesis

$$\Gamma((h(t/y))(\iota)) = \Gamma((h(\iota))(t/y)) \subseteq \Gamma((h(\iota)))(t/y)$$

Hence

$$\begin{aligned}
\Gamma(h(t/y)) &= \Delta(\mathrm{last}(h(t/y))) \cup \bigcup_{\iota < |\,\mathrm{last}(h)|} \Big(\Gamma((h(t/y))(\iota)) \setminus {\approx_{\mathbb{N}}} \Delta_\iota(\mathrm{last}(h(t/y)))\Big) \\
&\overset{i.h.}{\subseteq} \Delta(\mathrm{last}(h))(t/y) \cup \bigcup_{\iota < |\,\mathrm{last}(h)|} \Big(\Gamma((h)(\iota))(t/y) \setminus {\approx_{\mathbb{N}}} \Delta_\iota(\mathrm{last}(h))(t/y)\Big) \\
&\overset{!!!}{\subseteq} \Big(\Delta(\mathrm{last}(h)) \cup \bigcup_{\iota < |\,\mathrm{last}(h)|} \big(\Gamma((h)(\iota)) \setminus {\approx_{\mathbb{N}}} \Delta_\iota(\mathrm{last}(h))\big)\Big)(t/y) \\
&= \Gamma(h)(t/y)
\end{aligned}$$

This finishes the proof. ◻

We will now define the ingredients for a notation system for BA^∞, which forms the embedding of $\mathrm{BA}^\star$ into BA^∞.

Let $\mathcal{H}_{\mathrm{BA}}$ be the set of closed $\mathrm{BA}^\star$-derivations.

For each $h \in \mathcal{H}_{\mathrm{BA}}$ we define the denoted last inference $\mathrm{tp}(h)$ as follows: Let $h = \mathcal{I} h_0 \dots h_{n-1}$,

$$\mathrm{tp}(h) := \begin{cases}
\mathrm{Ax}_A & \text{if } \mathcal{I} = \mathrm{Ax}_\Delta, \text{ where } A \text{ is the “least” true literal in } \Delta \\
\bigwedge_{A_0 \wedge A_1} & \text{if } \mathcal{I} = \bigwedge_{A_0 \wedge A_1} \\
\bigvee^k_{A_0 \vee A_1} & \text{if } \mathcal{I} = \bigvee^k_{A_0 \vee A_1} \\
\bigwedge_{(\forall x)A} & \text{if } \mathcal{I} = \bigwedge^y_{(\forall x)A} \\
\bigvee^{t^{\mathbb{N}}}_{(\exists x)A} & \text{if } \mathcal{I} = \bigvee^t_{(\exists x)A} \\
\mathrm{Rep} & \text{if } \mathcal{I} = \mathrm{IND}^{y,t}_F \\
\mathrm{Rep} & \text{if } \mathcal{I} = \mathrm{IND}^{y,n,0}_F \\
\mathrm{Cut}_{F_y(\underline{n+2^i})} & \text{if } \mathcal{I} = \mathrm{IND}^{y,n,i+1}_F \\
\mathrm{Cut}_C & \text{if } \mathcal{I} = \mathrm{Cut}_C
\end{cases}$$

For each $h \in \mathcal{H}_{\mathrm{BA}}$ and $j \in \mathbb{N}$ we define the denoted sub-derivation $h[j]$ as follows: Let $h = \mathcal{I} h_0 \dots h_{n-1}$. If $j \geq |\,\mathrm{tp}(h)|$ let $h[j] := \mathrm{Ax}_{0=0}$. Otherwise, assume $j < |\,\mathrm{tp}(h)|$ and define

$$h[j] := \begin{cases}
h_{\min(j,1)} & \text{if } \mathcal{I} = \bigwedge_{A_0 \wedge A_1} \\
h_0 & \text{if } \mathcal{I} = \bigvee^k_{A_0 \vee A_1} \\
h_0(\underline{j}/y) & \text{if } \mathcal{I} = \bigwedge^y_{(\forall x)A} \\
h_0 & \text{if } \mathcal{I} = \bigvee^t_{(\exists x)A} \\
\mathrm{IND}^{y,0,|t|^{\mathbb{N}}}_F h_0 & \text{if } \mathcal{I} = \mathrm{IND}^{y,t}_F \\
h_0(\underline{n}/y) & \text{if } \mathcal{I} = \mathrm{IND}^{y,n,0}_F \\
\mathrm{IND}^{y,n,i}_F h_0 & \text{if } \mathcal{I} = \mathrm{IND}^{y,n,i+1}_F \text{ and } j = 0 \\
\mathrm{IND}^{y,n+2^i,i}_F h_0 & \text{if } \mathcal{I} = \mathrm{IND}^{y,n,i+1}_F \text{ and } j = 1 \\
h_j & \text{if } \mathcal{I} = \mathrm{Cut}_C
\end{cases}$$

The denoted end-sequent function on $\mathcal{H}_{\mathrm{BA}}$ is given by Γ computed according to Definition 2.4. The size function $|\cdot|$ on $\mathcal{H}_{\mathrm{BA}}$ is given by $|h| := \mathrm{sz}(h)$.

To define the denoted height function we need some analysis yielding an upper bound to the log of the lengths of inductions which may occur during the embedding (we take the log as this bounds the height of the derivation tree which embeds the application of induction). Let us first assume m is such an upper bound, and let us define the denoted height $\mathrm{o}_m(h)$ of h relative to m: For a $\mathrm{BA}^\star$-derivation $h = \mathcal{I}h_0 \dots h_{n-1}$ we define

$$\mathrm{o}_m(h) := \begin{cases} \mathrm{o}_m(h_0) + i + 1 & \text{if } \mathcal{I} = \mathrm{IND}_F^{y,n,i} \\ \mathrm{o}_m(h_0) + m + 1 & \text{if } \mathcal{I} = \mathrm{IND}_F^{y,t} \\ 1 + \sup_{i<n} \mathrm{o}_m(h_i) & \text{otherwise} \end{cases}$$

Observe that $\mathrm{o}_m(h) > 0$ (in particular, $\mathrm{o}(\mathrm{Ax}_\Delta) = 1$).

To fill the gap of providing a suitable upper bound function of $\mathrm{BA}^\star$-derivations we first need to fix monotone bounding terms for any term in $\mathcal{L}_{\mathrm{BA}}$.

Bounding terms

For a term t we define a term $\mathrm{bd}(t)$ which represents a monotone function with the following property: If $\mathrm{FV}(t) = \{\vec{x}\}$ then

$$(\forall \vec{n}) \qquad t_{\vec{x}}(\underline{\vec{n}})^{\mathbb{N}} \quad \leq \quad \mathrm{bd}(t)_{\vec{x}}(\underline{\vec{n}})^{\mathbb{N}}$$

Let $x_0, x_1, x_2, \dots$ be a fixed list of free variables. We fix for each function symbol f of arity n a *monotone bounding term* T_f with $\mathrm{FV}(T_f) \subseteq \{x_0, \dots, x_{n-1}\}$. E.g., assume that we have fixed for each function symbol f in our language a number $c_f \in \mathbb{N}$ such that $(\forall \vec{n})|f^{\mathbb{N}}(\vec{n})| \leq \max\{2, |\vec{n}|\}^{2^{c_f}}$ holds. We then can define

$$T_f \quad := \quad \underbrace{(\max\{2, \vec{x}\}) \# \dots \# (\max\{2, \vec{x}\})}_{2^{c_f} \text{ times}} \ .$$

As the only exception we demand that $T_{|\cdot|} := |x_0|$.

Now, let t be a term. If t is a closed term, let $\mathrm{bd}(t) := \underline{t^{\mathbb{N}}}$. If $t = ft_1 \dots t_n$ is not a closed term, let $\mathrm{bd}(t) := (T_f)_{\vec{x}}(\mathrm{bd}(t_1), \dots, \mathrm{bd}(t_n))$.

Bounding terms for $\mathrm{BA}^\star$-derivations

For $h \in \mathcal{H}_{\mathrm{BA}}$, the bounding term $\mathrm{bd}(h)$ is intended to bound any variable which occurs during the embedding of h, and the term $|\,\mathrm{ibd}(h)|$ is intended to bound the length of any induction which occurs during the embedding of h.

Let $h = \mathcal{I}h_0 \dots h_{n-1}$ be in $\mathcal{H}_{\mathrm{BA}}$.. We define

$$\mathrm{bd}(h) := \begin{cases} \max(\mathrm{bd}(h_0(\underline{\mathrm{bd}(t)}/y)), \mathrm{bd}(t)) & \text{if } \mathcal{I} = \bigwedge^y_{(\forall x \leq t)A} \\ \max(\mathrm{bd}(h_0), \mathrm{bd}(t)) & \text{if } \mathcal{I} = \bigvee^t_{(\exists x)A} \\ \max(\mathrm{bd}(h_0(\underline{2^{|\,\mathrm{bd}(t)|}}/y)), 2^{|\,\mathrm{bd}(t)|}) & \text{if } \mathcal{I} = \mathrm{IND}^{y,t}_F \\ \max(\mathrm{bd}(h_0(\underline{n+2^i}/y)), n+2^i) & \text{if } \mathcal{I} = \mathrm{IND}^{y,n,i}_F \\ \max(\mathrm{bd}(h_0), \dots, \mathrm{bd}(h_{n-1})) & \text{otherwise.} \end{cases}$$

$$\mathrm{ibd}(h) := \begin{cases} \mathrm{ibd}(h_0(\underline{\mathrm{bd}(t)}/y)) & \text{if } \mathcal{I} = \bigwedge^y_{(\forall x \leq t)A} \\ \max(\mathrm{ibd}(h_0(\underline{2^{|\,\mathrm{bd}(t)|}}/y)), 2^{|\,\mathrm{bd}(t)|}) & \text{if } \mathcal{I} = \mathrm{IND}^{y,t}_F \\ \max(\mathrm{ibd}(h_0(\underline{n+2^i}/y)), 2^i) & \text{if } \mathcal{I} = \mathrm{IND}^{y,n,i}_F \\ \max(\mathrm{ibd}(h_0), \dots, \mathrm{ibd}(h_{n-1})) & \text{otherwise.} \end{cases}$$

Now we can define the denoted height function $\mathrm{o}(h) := \mathrm{o}_{|\,\mathrm{ibd}(h)|}(h)$ for $h \in \mathcal{H}_{\mathrm{BA}}$.

Theorem 12.6. *The just defined system consisting of* $\mathcal{H}_{\mathrm{BA}}$*,* tp*,* $\cdot[\cdot]$*,* Γ*,* $\mathrm{o}(\cdot)$ *and* $|\cdot|$ *forms a notation system for* BA^∞ *in the sense of Definition 7.1.*

Proof. First, we observe that $\mathrm{o}(\cdot)$ satisfies the following monotonicity property:

$$m \leq m' \quad \Rightarrow \quad \mathrm{o}_m(h) \leq \mathrm{o}_{m'}(h) \ . \tag{2}$$

We also observe the following substitution property by inspection:

$$\mathrm{o}_m(h(t/y)) = \mathrm{o}_m(h) \ . \tag{3}$$

We prove the following slightly more general assertion:

$$m \geq |\,\mathrm{ibd}(h)| \quad \& \quad i < |\,\mathrm{tp}(h)| \quad \Rightarrow \quad \mathrm{o}_m(h[i]) < \mathrm{o}_m(h) \tag{4}$$

Then the assertion of the theorem follows using the monotonicity property (2), as $\mathrm{ibd}(h[i]) \leq \mathrm{ibd}(h)$.

The proof of (4) is by induction on the build-up of h. Let $h = \mathcal{I}h_0 \dots h_{n-1}$.

First assume that $h[i] = h_j(t/y)$. The definition of o_m immediately shows that in this case $\mathrm{o}_m(h) = 1 + \sup_{i<n} \mathrm{o}_m(h_i)$. The substitution property (3) shows that $\mathrm{o}_m(h_j(t/y)) = \mathrm{o}_m(h_j)$. Hence

$$\mathrm{o}_m(h) > \mathrm{o}_m(h_j) = \mathrm{o}_m(h_j(y/k)) = \mathrm{o}_m(h[i]) \ .$$

The remaining cases are the following ones:

If $h = \mathrm{IND}^{y,t}_F h_0$, then $h[0] = \mathrm{IND}^{y,0,|t|}_F h_0$. As $|t| \leq |\,\mathrm{bd}(t)| < |\,\mathrm{ibd}(h)| \leq m$ we obtain

$$\mathrm{o}_m(h[0]) = \mathrm{o}_m(h_0) + |t| + 1 < \mathrm{o}_m(h_0) + m + 1 = \mathrm{o}_m(h) \ .$$

If $h = \mathrm{IND}^{y,n,k+1}_F h_0$, then $h[i] = \mathrm{IND}^{y,n',k}_F h_0$ for some n' Hence

$$\mathrm{o}_m(h[i]) = \mathrm{o}_m(h_0) + k + 1 < \mathrm{o}_m(h_0) + k + 2 = \mathrm{o}_m(h) \ .$$

Thus, assertion (4) is proven. The Theorem follows using the next Proposition which shows the local faithfulness property of the denoted end-sequent function Γ. ◻

Proposition 12.7. Γ *satisfies the local faithfulness property: Let* $h \in \mathrm{BA}^\star$*, then*

$$\Delta(\mathrm{tp}(h)) \cup \bigcup_{\iota<|\,\mathrm{tp}(h)|} \Big(\Gamma(h[\iota]) \setminus \approx_{\mathbb{N}} \Delta_\iota(\mathrm{tp}(h))\Big) \subseteq \approx_{\mathbb{N}} \Gamma(h) \ .$$

Proof by induction on $\mathrm{o}(h)$. Let $h = \mathcal{I}h_0 \dots h_{n-1} \in \mathcal{H}_{\mathrm{BA}}$. We abbreviate

$$*(h) \quad := \quad \Delta(\mathrm{tp}(h)) \cup \bigcup_{\iota<|\,\mathrm{tp}(h)|} \Big(\Gamma(h[\iota]) \setminus \approx_{\mathbb{N}} \Delta_\iota(\mathrm{tp}(h))\Big) \ .$$

Case 1. $\mathcal{I} = \mathrm{Ax}_\Delta$: Let A be the "least" true literal in Δ, then

$$*(h) = \Delta(\mathrm{Ax}_A) = \{A\} \subseteq \Delta = \Gamma(h)$$

Case 2. $\mathcal{I} = \bigwedge_C$ for $C = A_0 \wedge A_1$: $\mathrm{tp}(h) = \bigwedge_C$, $h[0] = h_0$ and $C[0] = A_0$, and $h[\iota] = h_1$ and $C[\iota] = A_1$ for $\iota > 0$, hence

$$\begin{aligned} *(h) &= \{A_0 \ \wedge \ A_1\} \cup (\Gamma(h_0) \setminus \approx_{\mathbb{N}} \{A_0\}) \cup (\Gamma(h_1) \setminus \approx_{\mathbb{N}} \{A_1\}) \\ &= \Gamma(h) \end{aligned}$$

Case 3. $\mathcal{I} = \bigvee^k_{A_0 \vee A_1}$

Case 4. $\mathcal{I} = \bigwedge^y_{(\forall x)A}$: $\mathrm{tp}(h) = \bigwedge_{(\forall x)A}$, and $h[\iota] = h_0(\underline{\iota}/y)$ and $((\forall x)A)[\iota] = A(\underline{\iota}/x)$ for $\iota \in \mathbb{N}$ hence

$$\begin{aligned} *(h) &= \{(\forall x)A\} \cup \bigcup_{i\in\mathbb{N}} (\Gamma(h_0(\underline{i}/y)) \setminus \approx_{\mathbb{N}} \{A(\underline{i}/x)\}) \\ &\subseteq \{(\forall x)A\} \cup \bigcup_{i\in\mathbb{N}} (\Gamma(h_0)(\underline{i}/y) \setminus \approx_{\mathbb{N}} \{A(\underline{i}/x)\}) \\ &\overset{(1)}{=} \{(\forall x)A\} \cup \bigcup_{i\in\mathbb{N}} (\Gamma(h_0) \setminus \approx_{\mathbb{N}} \{A\}) \\ &= \Gamma(h) \end{aligned}$$

(1): uses Eigenvariable condition.

Case 5. $\mathcal{I} = \bigvee^t_{(\exists x)A}$: $\mathrm{tp}(h) = \bigvee^{t^{\mathbb{N}}}_{(\exists x)A}$ and $h[0] = h_0$, hence

$$\begin{aligned} *(h) &= \{(\exists x)A\} \cup (\Gamma(h_0) \setminus \approx_{\mathbb{N}} \{A(\underline{t}^{\mathbb{N}}/x)\}) \\ &= \{(\exists x)A\} \cup (\Gamma(h_0) \setminus \approx_{\mathbb{N}} \{A(t/x)\}) \\ &= \Gamma(h) \end{aligned}$$

Case 6. $\mathcal{I} = \mathrm{IND}_F^{y,t}$: $\mathrm{tp}(h) = \mathrm{Rep}$ and $h[0] = \mathrm{IND}_F^{y,0,|t^{\mathbb{N}}|} h_0$, hence

$$\begin{aligned}
*(h) &= \emptyset \cup \Gamma(\mathrm{IND}_F^{y,0,|t^{\mathbb{N}}|} h_0) \\
&= \{\neg F_y(\underline{0}), F_y(\underline{0 + 2^{|t^{\mathbb{N}}|}})\} \cup (\Gamma(h_0) \setminus \approx_{\mathbb{N}} \{\neg F, F_y(\mathrm{s}\, y)\}) \\
&\subseteq \approx_{\mathbb{N}} \{\neg F_y(0), F_y(2^{|t|})\} \cup (\Gamma(h_0) \setminus \approx_{\mathbb{N}} \{\neg F, F_y(\mathrm{s}\, y)\}) \\
&\subseteq \approx_{\mathbb{N}} \Gamma(h)
\end{aligned}$$

Case 7. $\mathcal{I} = \mathrm{IND}_F^{y,n,0}$: $\mathrm{tp}(h) = \mathrm{Rep}$ and $h[0] = h_0(\underline{n}/y)$, hence

$$\begin{aligned}
*(h) &= \emptyset \cup \Gamma(h_0(\underline{n}/y)) \\
&\subseteq \Gamma(h_0)(\underline{n}/y) \\
&\overset{(2)}{\subseteq} \approx_{\mathbb{N}} \{\neg F_y(\underline{n}), F_y(\mathrm{s}\, \underline{n})\} \cup (\Gamma(h_0) \setminus \approx_{\mathbb{N}} \{\neg F, F_y(\mathrm{s}\, y)\}) \\
&\subseteq \approx_{\mathbb{N}} \Big(\{\neg F_y(\underline{n}), F_y(\underline{n+1})\} \cup (\Gamma(h_0) \setminus \approx_{\mathbb{N}} \{\neg F, F_y(\mathrm{s}\, y)\}) \Big) \\
&= \approx_{\mathbb{N}} \Gamma(h)
\end{aligned}$$

(2) uses Eigenvariable condition.

Case 8. $\mathcal{I} = \mathrm{IND}_F^{y,n,i+1}$: $\mathrm{tp}(h) = \mathrm{Cut}_{F_y(\underline{n+2^i})}$, $h[0] = \mathrm{IND}_F^{y,n,i} h_0$, and $h[1] = \mathrm{IND}_F^{y,n+2^i,i} h_0$, hence (abbreviating $\Xi := \Gamma(h_0) \setminus \approx_{\mathbb{N}} \{\neg F, F_y(\mathrm{s}\, y)\}$)

$$\begin{aligned}
*(h) &= \emptyset \cup \Big(\Gamma(\mathrm{IND}_F^{y,n,i} h_0) \setminus \approx_{\mathbb{N}} \{F_y(\underline{n+2^i})\} \Big) \\
&\quad \cup \Big(\Gamma(\mathrm{IND}_F^{y,n+2^i,i} h_0) \setminus \approx_{\mathbb{N}} \{\neg F_y(\underline{n+2^i})\} \Big) \\
&= \Big(\big(\{\neg F_y(\underline{n}), F_y(\underline{n+2^i})\} \cup \Xi \big) \setminus \approx_{\mathbb{N}} \{F_y(\underline{n+2^i})\} \Big) \\
&\quad \cup \Big(\big(\{\neg F_y(\underline{n+2^i}), F_y(\underline{n+2^{i+1}})\} \cup \Xi \big) \setminus \approx_{\mathbb{N}} \{\neg F_y(\underline{n+2^i})\} \Big) \\
&\subseteq \{\neg F_y(\underline{n}), F_y(\underline{n+2^{i+1}})\} \cup \Xi \\
&= \Gamma(h)
\end{aligned}$$

Case 9. $\mathcal{I} = \mathrm{Cut}_C$: $\mathrm{tp}(h) = \mathrm{Cut}_C$ and $h[\iota] = h_\iota$ for $\iota < 2$, hence

$$\begin{aligned}
*(h) &= \emptyset \cup (\Gamma(h_0) \setminus \approx_{\mathbb{N}} \{C\}) \cup (\Gamma(h_1) \setminus \approx_{\mathbb{N}} \{\neg C\}) \\
&= \Gamma(h)
\end{aligned}$$

∎

Observation 12.8. *The following relations and functions are polynomial time computable: the finitary proof system* $\mathrm{BA}^\star$*, the set of* $\mathrm{BA}^\star$*-quasi derivations and the functions* $h \mapsto \Gamma(h)$*,* $h \mapsto \mathrm{hgt}(h)$*, and* $h \mapsto \mathrm{sz}(h)$ *denoting the endsequent, the height and the size for a* $\mathrm{BA}^\star$*-quasi derivation* h*; the bounding term* $t \mapsto \mathrm{bd}(t)$ *for terms* t *occurring in* $\mathcal{F}_{\mathrm{BA}}$ *and the relations* $\mathrm{bd}(h) \leq m$ *and* $\mathrm{ibd}(h) \leq m$ *on* $\mathcal{H}_{\mathrm{BA}} \times \mathbb{N}$*; the set* $\mathcal{H}_{\mathrm{BA}}$ *and the functions* $h \mapsto \mathrm{tp}(h)$*,* $h, i \mapsto h[i]$*,* $h \mapsto \Gamma(h)$*,* $m, h \mapsto \mathrm{o}_m(h)$ *and* $h \mapsto |h|$*.*

Proof. For bounding terms we use our assumption that a fixed (finite) number of function symbols and term depth is only allowed, which implies that terms can only denote a fixed finite number of different polynomial time computable functions. That $\mathrm{bd}(h) \leq m$ is polynomial time computable is clear as the computation of $\mathrm{bd}(h)$ computes a monotone increasing sequence of values by successively applying one of the finitely many polynomial time computable functions, and once the bound m is exceeded during this process we can already output *NO*. $\square$

As the function $\mathrm{bd}(h)$ in general may not be polynomially bounded, we cannot conclude in general that $\mathrm{o}(h)$ is polynomial time computable. However, the function $m, h \mapsto \mathrm{o}_{\min(|\,\mathrm{ibd}(h)|,m)}(h)$ is polynomial time computable and will be sufficient in our applications.

13 Computational content of proofs

Let us start by describing the idea for computing witnesses using proof trees. Assume we have a BA proof of an existential formula $(\exists y)\varphi(y)$ and we want to compute a k such that $\varphi(k)$ is true – in case we are interested in definable functions, such a situation is obtained from a proof of $(\forall x)(\exists y)\varphi(x, y)$ by inverting the universal quantifier to some $n \in \mathbb{N}$. Assume further, we have applied some proof theoretical transformations to obtain a BA^∞ derivation d of $(\exists y)\varphi(y)$ with $\mathcal{C}$-$\mathrm{crk}(d) \leq \mathcal{C}$-$\mathrm{rk}(\varphi)$ for some set of formulae $\mathcal{C}$ (the choice of $\mathcal{C}$ depends on the level of definability we are interested in). Then we can define a path through d, represented by sub-derivations

$$d = d_0, d_1, d_2, \ldots$$

with

- $d_{\ell+1} = d_\ell(i)$ for some $i \in |\,\mathrm{last}(d_\ell)|$
- $\Gamma(d_\ell) = (\exists y)\varphi(y), \Gamma_\ell$ where all formulae $A \in \Gamma_\ell$ are false and satisfy $\mathcal{C}$-$\mathrm{rk}(A) \leq \mathcal{C}$-$\mathrm{rk}(\varphi)$.

As d is well-founded, such a path must be finite, i.e. ends with some d_ℓ say. In this situation we must have that $\mathrm{last}(d_\ell) = \bigvee^k_{(\exists y)\varphi(y)}$ and that $\varphi(k)$ is true. Hence we can output k.

Such a path can be viewed as the canonical path to the following local search problem: Let F be a set of possible solutions, which is a subset of BA^∞ containing only those d' which satisfy that $\Gamma(d') \subseteq \{(\exists y)\varphi(y)\} \cup \Gamma'$ where all formulae $A \in \Gamma'$ are false and satisfy $\mathcal{C}$-$\mathrm{rk}(A) \leq \mathcal{C}$-$\mathrm{rk}(\varphi)$. Furthermore, assume $d \in F$ and that F is closed under the following neighbourhood function $N\colon \mathrm{BA}^\infty \to \mathrm{BA}^\infty$ which is defined by case distinction on the shape of $\mathrm{last}(d')$ for $d' \in F$:

- $\mathrm{last}(d) = \mathrm{Ax}_A$ cannot occur as all atomic formulae in $\Gamma(d')$ are false.

- $\mathrm{last}(d) = \bigwedge_{A_0 \wedge A_1}$, then $A_0 \wedge A_1$ must be false, hence some of A_0, A_1 must be false. Let $N(d) := d(0)$ if A_0 is false, and $d(1)$ otherwise.

- $\mathrm{last}(d) = \bigwedge_{A_0 \vee A_1}$, then $A_0 \vee A_1$ must be false, hence both A_0, A_1 must be false. Let $N(d) := d(0)$.

- $\mathrm{last}(d) = \bigwedge_{(\forall x)A(x)}$. As $(\forall x)A(x)$ is false there is some i such that $A(i)$ is false. Let $N(d) := d(i)$.

- $\mathrm{last}(d) = \bigvee^k_{(\exists x)A(x)}$. If $(\exists x)A(x)$ is different from $(\exists y)\varphi(y)$ then $(\exists x)A(x)$ must be false; let $N(d) := d(0)$. Otherwise, let $N(d) = d(0)$ in case $\varphi(k)$ is false, and $N(d) = d$ in case it is true (in which case we found a true solution to the original search problem).

- $\mathrm{last}(d_\ell) = \mathrm{Cut}_C$. If C is false let $N(d) := d(0)$, otherwise let $N(d) := d(1)$.

The idea in the following will be to use proof notations from $\mathcal{H}_{\mathrm{BA}}$ to denote this search problem. This way we will obtain characterisations of the definable functions of Bounded Arithmetic theories.

The level of proof theoretic reduction will be adjusted in such a way that occurring formulae which have to be decided fall exactly in the computational class under consideration. So our main concern in order for this strategy to be meaningful is to find feasible upper bounds for the length of such reduction sequences and for the complexity of derivation notations occurring in them.

13.1 Complexity notions for $\mathrm{BA}^\star$

In order to handle the complexity of $\mathrm{BA}^\star$ proof notations occurring in the set of possible solutions, we need some notions describing key complexity properties of them which we will provide first.

Although $\mathrm{tp}(A) = \bigwedge$ for any A starting with a $\forall$, and thus we can denote infinitely many direct sub-formulae by $A[n]$ for all $n \in \mathbb{N}$, only finitely many carry non-trivial information, because all quantifiers (and in particular this outermost $\forall$) are bounded. The next definition makes this formal by assigning first to each closed formula in $\mathcal{F}_{\mathrm{BA}}$, then to each inference symbol in BA^∞, and finally to each proof notation in $\mathcal{CH}_{\mathrm{BA}}$, its range.

Definition 13.1. Let A be a formula in $\mathcal{F}_{\mathrm{BA}}$. We define *the range of A,* denoted $\mathrm{rng}(A)$, by

$$\mathrm{rng}(A) := \begin{cases} 0 & \text{if } A \text{ a literal },\\ 2 & \text{if } A = B \wedge C \text{ or } A = B \vee C \ ,\\ t^{\mathbb{N}} + 1 & \text{if } A = (\forall x \le t)B \text{ or } A = (\exists x \le t)B \ . \end{cases}$$

Let $\mathcal{I}$ be an inference symbol of BA^∞. We define *the range of $\mathcal{I}$,* denoted

$\mathrm{rng}(\mathcal{I})$, by

$$\mathrm{rng}(\mathcal{I}) := \begin{cases} 0 & \text{if } \mathcal{I} = \mathrm{Ax}_A \ , \\ 1 & \text{if } \mathcal{I} = \bigvee_C^k \text{ or } \mathcal{I} = \mathrm{Rep} \ , \\ \mathrm{rng}(C) & \text{if } \mathcal{I} = \bigwedge_C \ , \\ 2 & \text{if } \mathcal{I} = \mathrm{Cut}_C \ . \end{cases}$$

For $h \in \mathcal{CH}_{\mathrm{BA}}$ we define

$$\mathrm{rng}(h) := \mathrm{rng}(\mathrm{tp}(h)) \ .$$

Definition 13.2. We extend the definition of bounding terms $\mathrm{bd}(h)$ and $\mathrm{ibd}(h)$ from $\mathcal{H}_{\mathrm{BA}}$ to $\mathcal{CH}_{\mathrm{BA}}$ in the following way by induction on the build-up of $h \in \mathcal{CH}_{\mathrm{BA}}$:

- If $h \in \mathcal{H}_{\mathrm{BA}}$ then the definition of $\mathrm{bd}(h)$ and $\mathrm{ibd}(h)$ are inherited from the definition of bd resp. $\mathrm{ibd}(h)$ on $\mathcal{H}_{\mathrm{BA}}$.
- If $h = \mathsf{I}_C^k h_0$ then

$$\mathrm{bd}(h) := \begin{cases} \mathrm{bd}(h_0) & \text{if } k < \mathrm{rng}(C) \ , \\ 0 & \text{otherwise} \ . \end{cases}$$
$$\mathrm{ibd}(h) := \mathrm{ibd}(h_0)$$

- $\mathrm{bd}(\mathsf{R}_C h_0 h_1) := \max\{\mathrm{bd}(h_0), \mathrm{bd}(h_1)\}$, $\mathrm{ibd}(\mathsf{R}_C h_0 h_1) := \max\{\mathrm{ibd}(h_0), \mathrm{ibd}(h_1)\}$.
- $\mathrm{bd}(\mathsf{E} h_0) := \mathrm{bd}(h_0)$, $\mathrm{ibd}(\mathsf{E} h_0) := \mathrm{ibd}(h_0)$.

Lemma 13.3. *Let $h \in \mathcal{CH}_{\mathrm{BA}}$.*

1. *If $j < \mathrm{rng}(h)$ then $\mathrm{bd}(h[j]) \leq \mathrm{bd}(h)$ and $\mathrm{ibd}(h[j]) \leq \mathrm{ibd}(h)$.*
2. *If $\mathrm{tp}(h) = \bigvee_C^k$ then $k \leq \mathrm{bd}(h)$.*

Proof by induction on the build-up of h. ◻

Definition 13.4. For $h \in \mathrm{BA}^\star \cup \mathcal{CH}_{\mathrm{BA}}$ we define *the set of decorations of h*, $\mathrm{deco}(h) \in \mathfrak{P}_{fin}(\mathrm{BFOR})$, by induction on the build-up of h. Let $h = \mathcal{I} h_0 \dots h_{n-1}$. We define

$$\mathrm{deco}(h) := \mathrm{deco}(\mathcal{I})\Big(\bigcup_{i<n} \mathrm{deco}(h_i)\Big)$$

where

$$\mathrm{deco}(\mathcal{I})(S) := \begin{cases} S \cup \Delta(\mathcal{I}) \cup \{F\} & \text{if } \mathcal{I} = \mathrm{IND}_F^{y,t} \text{ or } \mathcal{I} = \mathrm{IND}_F^{y,a,i} \ , \\ S \cup \Delta(\mathcal{I}) & \text{otherwise} \ . \end{cases}$$

Observation 13.5. *We have $\Gamma(h) \subseteq \mathrm{deco}(h)$.*

Definition 13.6. Let $\mathcal{C}\mathrm{omp}\mathcal{H}_{\mathrm{BA}}$ be the set of all $h \in \mathcal{CH}_{\mathrm{BA}}$ which have the property that all occurrences of I_C^k in h satisfy $k < \mathrm{rng}(C)$.

Definition 13.7. Let Φ be a finite set of formulae in BFOR, and let $K \in \mathbb{N}$ be a size parameter. With Φ_K we denote the set of formulae which result from formulae in Φ by substituting free variables by constants from $\{c_i : 0 \leq i \leq K\}$.

Lemma 13.8. *Let $h \in \mathcal{C}\mathrm{omp}\mathcal{H}_{\mathrm{BA}}$ and $\Phi \in \mathfrak{P}_{fin}(\mathrm{BFOR})$ such that $\mathrm{deco}(h) \subseteq \Phi$, and Φ is closed under negation and taking sub-formulae. Let $j, K \in \mathbb{N}$ and y be a variable.*

1. *If $j \leq K$ and $C \in \Phi$, then $C[j] \in \Phi_K$.*
2. *If $j \leq K$ then $\mathrm{deco}(h(\underline{j}/y)) \subseteq \Phi_K$.*
3. *$\Delta(\mathrm{tp}(h)) \subseteq \mathrm{deco}(h)_{\mathrm{bd}(h)}$ (subscript $\mathrm{bd}(h)$ needed e.g. for $\mathrm{IND}_F^{y,n,i+1}$).*
4. *If $j < \mathrm{rng}(h)$ then $\mathrm{deco}(h[j]) \subseteq \Phi_{\mathrm{bd}(h)}$.*

Proof. For 4., consider the case that $h = \mathsf{R}_C h_0 h_1$, $\mathrm{tp}(h_1) = \bigvee_{\neg C}^{k}$ and $j = 0$, i.e. $h[0] = \mathsf{I}_C^k h_0$. By 3. we have $\neg C \in \Phi_{\mathrm{bd}(h_1)}$, hence $C \in \Phi_{\mathrm{bd}(h)}$. Also $k \leq \mathrm{bd}(h_1)$ by Lemma 13.3, 2. Hence, $C[k] \in \Phi_{\mathrm{bd}(h)}$ by 1. Now we compute

$$\mathrm{deco}(h[0]) = \{C[k]\} \cup \mathrm{deco}(h_0) \subseteq \Phi_{\mathrm{bd}(h)} \cup \Phi = \Phi_{\mathrm{bd}(h)} \quad .$$

□

Lemma 13.9. *For $h \in \mathcal{CH}_{\mathrm{BA}}$ we have that the cardinality of $\Gamma(h)$ is bounded above by $2 \cdot \mathrm{sz}(h)$.*

Proof. Let the cardinality of a set S be denoted by $\mathrm{card}(S)$. We observe that $\mathrm{card}(\Delta(\mathcal{I})) \leq 2$ for any $\mathcal{I} \in \mathrm{BA}^\infty$. Thus we can compute for $h = \mathcal{I} h_0 \dots h_{n-1} \in \mathcal{CH}_{\mathrm{BA}}$

$$\mathrm{card}(\Gamma(h)) \leq \mathrm{card}(\Delta(\mathcal{I})) + \sum_{i<n} \mathrm{card}(\Gamma(h_i)) \leq 2 + \sum_{i<n} 2 \cdot \mathrm{sz}(h_i) = 2 \cdot \mathrm{sz}(h) \quad .$$

□

13.2 Search problems defined by proof notations

We identify the notation system $\mathcal{H}_{\mathrm{BA}}$ for BA^∞ with the abstract system of proof notations associated with it according to Observation 8.2. For $s \in \mathbb{N}$ a size parameter we define

$$\mathcal{H}_{\mathrm{BA}}^s := \{h \in \mathcal{H}_{\mathrm{BA}} : |h| \leq s\} \quad .$$

Then $\mathcal{H}_{\mathrm{BA}}^s$ is an s-bounded, abstract system of proof notations, because we observe that $h \in \mathcal{H}_{\mathrm{BA}}$ and $h \to h'$ implies $|h'| \leq |h|$.

Remember that $\overline{h}$ for $h \in \mathcal{CH}_{\mathrm{BA}}$ denotes the abstraction of h which allows us to view $\mathcal{CH}_{\mathrm{BA}}$ as a subsystem of $\widetilde{\mathcal{H}_{\mathrm{BA}}}$ (see Definition 8.4 and Observation 8.5).

Definition 13.10. For $h \in \mathcal{CH}_{\mathrm{BA}}$ we define $\vartheta(h)(s) := \vartheta(\overline{h})(s)$.

Theorem 9.7 now reads as follows:

Corollary 13.11. *If $h \in \mathcal{CH}^s_{\mathrm{BA}}$ and $h \to h'$, then $\vartheta(h)(s) \geq \vartheta(h')(s)$.*

Definition 13.12. We define a local search problem L parameterised by

- a finite set of bounded formulae $\Phi \subset \mathrm{BFOR}$,
- a *"complexity class"* $\mathcal{C}$ given as a polynomial time computable set of $\mathcal{L}_{\mathrm{BA}}$-formulae (usually $\mathcal{C} = \Sigma^{\mathrm{b}}_i$ for some i),
- a *size parameter* $s \in \mathbb{N}$,
- an *initial value function* $h_\cdot \colon \mathbb{N} \to \mathcal{C}\mathrm{omp}\mathcal{H}^s_{\mathrm{BA}}$, where h_a is presented in the form $\mathsf{E}\ldots\mathsf{E}h(\underline{a}/x)$ for some $\mathrm{BA}^\star$-derivation h,
- a formula $(\exists y)\varphi(x,y) \in \Phi$ with $\neg\varphi \in \mathcal{C}$,

such that, for $a \in \mathbb{N}$,

- $\Gamma(h_a) = \{(\exists y)\varphi(\underline{a}, y)\}$,
- $\mathcal{C}\text{-}\mathrm{crk}(h_a) \leq 1$,
- $\mathrm{o}(h_a) = 2^{|a|^{O(1)}}$,
- $\vartheta(h_a)(s) = |a|^{O(1)}$,
- $\mathrm{deco}(h_a) \subseteq \Phi_a$,

in the following way:

- The set of *possible solutions* $F(a) \in \mathfrak{P}_{fin}(\mathcal{C}\mathrm{omp}\mathcal{H}^s_{\mathrm{BA}})$ is given as the set of those $h \in \mathcal{C}\mathrm{omp}\mathcal{H}^s_{\mathrm{BA}}$ which satisfy:
  i) $\Gamma(h) \subseteq \{(\exists y)\varphi(\underline{a}, y)\} \cup \Delta$ for some $\Delta \subseteq \mathcal{C} \cup \neg\mathcal{C}$ such that all $A \in \Delta$ are closed and false,
  ii) $\mathcal{C}\text{-}\mathrm{crk}(h) \leq 1$,
  iii) $\mathrm{o}(h) \leq \mathrm{o}(h_a)$,
  iv) $\vartheta(h)(s) \leq \vartheta(h_a)(s)$,
  v) $\mathrm{bd}(h) \leq \mathrm{bd}(h_a)$ and $\mathrm{ibd}(h) \leq \mathrm{ibd}(h_a)$,
  vi) $\mathrm{deco}(h) \subseteq \Phi_{\mathrm{bd}(h_a)}$;
- The *initial value function* is given by $i(a) := h_a$;
- the *cost function* is defined as $c(a,h) := \mathrm{o}(h)$; and

- the *neighbourhood function* is given by

$$N(a,h) := \begin{cases} h[j] & \text{if } \operatorname{tp}(h) = \bigwedge_C, j < \operatorname{rng}(C) \text{ and } C[j] \text{ false },\\ h[0] & \text{if } \operatorname{tp}(h) = \bigvee_C^i \text{ and } C \neq (\exists y)\varphi(\underline{a}, y)\\ & \quad\text{or } \operatorname{tp}(h) = \bigvee^i_{(\exists y)\varphi(\underline{a},y)} \text{ and } \varphi(\underline{a}, \underline{i}) \text{ false },\\ h[0] & \text{if } \operatorname{tp}(h) = \operatorname{Cut}_C \text{ and } C \text{ false },\\ h[1] & \text{if } \operatorname{tp}(h) = \operatorname{Cut}_C \text{ and } C \text{ true },\\ h[0] & \text{if } \operatorname{tp}(h) = \operatorname{Rep},\\ h & \text{otherwise}. \end{cases}$$

(Observe that the just defined neighbourhood function is a multi-function due to case $\bigwedge_C$.)

Proof. First observe that the initial value is indeed a possible solution, $i(a) = h_a \in F(a)$.

Let $h \in F(a)$, $h' := N(a,h)$. Then we show

1. $h \neq h'$ implies $h \to h'$ and $\mathrm{o}(h') < \mathrm{o}(h)$,
2. $h' \in F(a)$.

For $h = h'$ the assertions are obvious. So let us assume $h \neq h'$. Then $h' = h[j]$ for some $j < \operatorname{rng}(h)$ by construction. Hence, the first claim is obvious.

For the second claim, we consider i)–vi) of the definition of $h' \in F(a)$: ii) is clear; iii) is obvious; for iv) observe that $h \to h'$, thus $\vartheta(h')(s) \leq \vartheta(h)(s)$ by Corollary 13.11; for v) observe that $j < \operatorname{rng}(h)$ implies $\operatorname{bd}(h') \leq \operatorname{bd}(h)$ and $\operatorname{ibd}(h') \leq \operatorname{ibd}(h)$ by Lemma 13.3; for vi) observe that $j < \operatorname{rng}(h)$ implies $\operatorname{deco}(h') \subseteq (\Phi_{\operatorname{bd}(h_a)})_{\operatorname{bd}(h)} = \Phi_{\operatorname{bd}(h_a)}$ by Lemma 13.8, 2., because $\operatorname{bd}(h) \leq \operatorname{bd}(h_a)$. And finally for i) we first observe that the first condition that $\Gamma(h) \setminus \{(\exists y)\varphi(\underline{a}, y)\}$ is a subset of $\mathcal{C} \cup \neg\mathcal{C}$ consisting only of closed formulae, is satisfied, as $\mathcal{C}\text{-}\operatorname{crk}(h) \leq 1$. For the second condition of i) let $\mathcal{I} := \operatorname{tp}(h)$. We have by Proposition 7.2 that

$$\Gamma(h[j]) \subseteq \approx_{\mathbb{N}}\Big(\Gamma(h) \cup \Delta_j(\mathcal{I})\Big)$$

thus it is enough to show that $\bigvee \Delta_j(\mathcal{I})$ is false.

- $\mathcal{I} = \bigwedge_C$: $\Delta_j(\mathcal{I}) = \{C[j]\}$ and $C[j]$ false by construction.
- $\mathcal{I} = \bigvee_C^i$: then $j = 0$. If $C \neq (\exists y)\varphi(\underline{a}, y)$, then $\Delta_0(\mathcal{I}) = \{C[i]\}$. Now C is false by i) of $h \in F(a)$, hence $C[i]$ must be false as well. Otherwise, $\Delta_0(\mathcal{I}) = \{\varphi(\underline{a}, \underline{i})\}$, and $\varphi(\underline{a}, \underline{i})$ false by construction.
- $\mathcal{I} = \operatorname{Cut}_C$: If $j = 0$, then $\Delta_0(\mathcal{I}) = \{C\}$ and C false by construction. Otherwise, $j = 1$, then $\Delta_1(\mathcal{I}) = \{\neg C\}$ and $\neg C$ false by construction.
- $\mathcal{I} = \operatorname{Rep}$: then $j = 0$ and $\Delta_0(\mathcal{I}) = \emptyset$ and nothing is to show.

∎

Proposition 13.13 (Complexity of L)**.** *$F \in \mathrm{P}^{\mathcal{C}}$, $i, c \in \mathrm{FP}$, and $N \in \mathrm{FP}^{\mathcal{C}}[wit, 1]$.*

Proof. First observe that the functions $a \mapsto i(a) = h_a$, $a \mapsto \mathrm{bd}(h_a)$, $a \mapsto \mathrm{ibd}(h_a)$, $a \mapsto \mathrm{o}(h_a)$, $a \mapsto \vartheta(h_a)$, and $a \mapsto \mathrm{deco}(h_a)$ are polynomial time computable.

Furthermore, the relations $\mathcal{C}\mathrm{omp}\mathcal{H}^s_{\mathrm{BA}}$, $\mathcal{C}$-$\mathrm{crk}(h) \leq 1$, $\mathrm{bd}(h) \leq m$, $\mathrm{ibd}(h) \leq m$ and $\mathrm{deco}(h) \subseteq \Phi_m$ are polynomial time computable, and once $\mathrm{ibd}(h) \leq m$ is established we also can compute $\mathrm{o}(h) \leq m'$ and then $\mathrm{o}(h)$ in polynomial time. Hence $c \in \mathrm{FP}$

Also, the functions $\mathrm{tp}(h)$ and $h[i]$ are polynomial time computable on $\mathcal{CH}_{\mathrm{BA}}$, which shows $N \in \mathrm{FP}^{\mathcal{C}}[\mathrm{wit}, 1]$.

For $F \in \mathrm{P}^{\mathcal{C}}$ observe that $\Gamma(h) \subseteq \mathrm{deco}(h) \subseteq \Phi_{\mathrm{bd}(h_a)}$, hence condition $h \in F(a)$, i), is a property in $\mathrm{P}^{\mathcal{C}}$. ∎

Proposition 13.14 (Properties of L)**.**

1. *$N(a, h) = h$ implies $\mathrm{tp}(h) = \bigvee^i_{(\exists y)\varphi(\underline{a}, y)}$ with $\varphi(\underline{a}, \underline{i})$ true. Thus, the local search problem L defines a multi-function by mapping a to i (this is called the computed multi-function).*

2. *The search problem L in general defines a search problem in $PLS^{\mathcal{C}}$, assuming that we turn the neighbourhood (multi-)function into a real function, which can easily be achieved by using an intermediate $PLS^{\mathcal{C}}$ search problem which looks for the smallest witness for the case $\mathrm{tp}(h) = \bigwedge_C$. Then $N \in \mathrm{FP}^{\mathcal{C}}$.*

3. *Assume $\mathrm{o}(h_a) = |a|^{O(1)}$. Then the canonical path through L, which starts at h_a and leads to a local minimum, is of polynomial length with terms of polynomial size, thus the computed multi-function is in $\mathrm{FP}^{\mathcal{C}}[wit, \mathrm{o}(h_a)]$.*

∎

13.3 Σ^{b}_i-definable multi-functions in S^{i-1}_2

Let $i \geq 2$ and assume that $\mathrm{S}^{i-1}_2 \vdash (\forall x)(\exists y)\varphi(x, y)$ with $(\exists y)\varphi(x, y) \in \Sigma^{\mathrm{b}}_i$, $\varphi \in \Pi^{\mathrm{b}}_{i-1}$. By partial cut-elimination we obtain some $\mathrm{BA}^\star$-derivation h such that

- $\mathrm{FV}(h) \subseteq \{x\}$,
- $\Gamma(h) = \{(\exists y)\varphi(x, y)\}$,
- $\Sigma^{\mathrm{b}}_{i-1}$-$\mathrm{crk}(h) \leq 1$, and
- $\mathrm{o}(h(\underline{a}/x)) = O(||a||)$.

We define a search problem by stating its parameters:

- $\Phi := \mathrm{deco}(h)$ is a finite set of formulae in BFOR,
- as the "complexity class" we take $\mathcal{C} := \Sigma^{\mathrm{b}}_{i-1}$,
- for the size parameter we choose $s := |h|$,

- the initial value function is given by $h_a := h(\underline{a}/x)$,
- the formula is as given, $(\exists y)\varphi(x, y)$.

This defines a local search problem according to Definition 13.12, because

- $\Gamma(h_a) = \Gamma(h(\underline{a}/x)) = \Gamma(h)(\underline{a}/x) = \{(\exists y)\varphi(\underline{a}, y)\}$,
- as $h \in \mathcal{H}^s_{\mathrm{BA}}$ we have $h(\underline{a}/x) \in \mathcal{H}^s_{\mathrm{BA}}$, hence $\vartheta(h_a)(s) = s = O(1)$
- $\mathrm{deco}(h_a) \subseteq \Phi_a$ by Lemma 13.8, 1.

As $\mathrm{o}(h_a) = O(||a||)$, Proposition 13.14, 3., shows that the computed multi-function of this search problem is in $\mathrm{FP}^{\Sigma^{\mathrm{b}}_{i-1}}[\mathrm{wit}, O(\log n)]$, which coincides with the description given by Krajíček [Kra93].

13.4 Σ^{b}_i-definable functions in S^i_2

Let $i > 0$ and assume that $\mathrm{S}^i_2 \vdash (\forall x)(\exists y)\varphi(x, y)$ with $(\exists y)\varphi(x, y) \in \Sigma^{\mathrm{b}}_i$, $\varphi \in \Pi^{\mathrm{b}}_{i-1}$. By partial cut-elimination we obtain some $\mathrm{BA}^\star$-derivation h such that

- $\mathrm{FV}(h) \subseteq \{x\}$,
- $\Gamma(h) = \{(\exists y)\varphi(x, y)\}$,
- $\Sigma^{\mathrm{b}}_{i-1}\text{-crk}(h) \leq 2$, and
- $\mathrm{o}(h(\underline{a}/x)) = O(||a||)$.

We define a search problem by stating its parameters:

- $\Phi := \mathrm{deco}(h)$ is a finite set of formulae in BFOR,
- as the "complexity class" we take $\mathcal{C} := \Sigma^{\mathrm{b}}_{i-1}$,
- for the size parameter we choose $s := |h|$,
- the initial value function is given by $h_a := \mathsf{E}h(\underline{a}/x)$,
- the formula is as given, $(\exists y)\varphi(x, y)$.

This defines a local search problem according to Definition 13.12, because

- $\Gamma(h_a) = \{(\exists y)\varphi(\underline{a}, y)\}$,
- $\Sigma^{\mathrm{b}}_{i-1}\text{-crk}(h_a) \leq 1$,
- $\mathrm{o}(h_a) = 2^{\mathrm{o}(h(\underline{a}/x))} - 1 = 2^{O(||a||)} = |a|^{O(1)}$,
- as $h(\underline{a}/x) \in \mathcal{H}^s_{\mathrm{BA}}$ we have

$$\begin{aligned}
\vartheta(h_a)(s) &= \vartheta(\mathsf{E}h(\underline{a}/x))(s) \\
&= \mathrm{o}(h(\underline{a}/x)) \cdot (\vartheta(h(\underline{a}/x))(s) + 2) \\
&= O(||a||) \cdot (s + 2) = O(||a||)
\end{aligned}$$

- $\mathrm{deco}(h_a) \subseteq \Phi_a$.

As $\mathrm{o}(h_a) = |a|^{O(1)}$, Proposition 13.14, 3., shows that the computed multi-function of this search problem is in $\mathrm{FP}^{\Sigma^{\mathrm{b}}_{i-1}}[\mathrm{wit}, n^{O(1)}] = \mathrm{FP}^{\Sigma^{\mathrm{b}}_{i-1}}[\mathrm{wit}]$.

But this immediately implies that the Σ^{b}_i-definable functions of S^i_2 are in $\mathrm{FP}^{\Sigma^{\mathrm{b}}_{i-1}}$, because a witness query to $(\exists z < t)\psi(u,z)$ can be replaced by $|t|$ many usual (non-witness) queries to $\chi(a,b,u) = (\exists z < t)(a \leq z < b \wedge \psi(u,z))$ using a divide and conquer strategy. This characterisation coincides with the one given by Buss [Bus86].

13.5 Σ^{b}_i-definable multi-functions in S^{i+1}_2

Let $i > 0$ and assume that $\mathrm{S}^{i+1}_2 \vdash (\forall x)(\exists y)\varphi(x,y)$ with $(\exists y)\varphi(x,y) \in \Sigma^{\mathrm{b}}_i$, $\varphi \in \Pi^{\mathrm{b}}_{i-1}$. By partial cut-elimination we obtain some $\mathrm{BA}^\star$-derivation h such that

- $\mathrm{FV}(h) \subseteq \{x\}$,
- $\Gamma(h) = \{(\exists y)\varphi(x,y)\}$,
- $\Sigma^{\mathrm{b}}_{i-1}$-$\mathrm{crk}(h) \leq 3$, and
- $\mathrm{o}(h(\underline{a}/x)) = O(||a||)$.

We define a search problem by stating its parameters:

- $\Phi := \mathrm{deco}(h)$ is a finite set of formulae in BFOR,
- as the "complexity class" we take $\mathcal{C} := \Sigma^{\mathrm{b}}_{i-1}$,
- for the size parameter we choose $s := |h|$,
- the initial value function is given by $h_a := \mathsf{EE}h(\underline{a}/x)$,
- the formula is as given, $(\exists y)\varphi(x,y)$.

This defines a local search problem according to Definition 13.12, because

- $\Gamma(h_a) = \{(\exists y)\varphi(\underline{a},y)\}$,
- $\Sigma^{\mathrm{b}}_{i-1}$-$\mathrm{crk}(h_a) \leq 1$,
- $\mathrm{o}(h_a) = 2^{\mathrm{o}(\mathsf{E}h(\underline{a}/x))} - 1 = 2^{|a|^{O(1)}}$,
- as $h(\underline{a}/x) \in \mathcal{H}^s_{\mathrm{BA}}$ we have

$$\begin{aligned}\vartheta(h_a)(s) &= \vartheta(\mathsf{EE}h(\underline{a}/x))(s)\\ &= \mathrm{o}(\mathsf{E}h(\underline{a}/x)) \cdot (\vartheta(\mathsf{E}h(\underline{a}/x))(s) + 2)\\ &= |a|^{O(1)} \cdot (O(||a||) + 2) = |a|^{O(1)}\end{aligned}$$

- $\mathrm{deco}(h_a) \subseteq \Phi_a$.

By Proposition 13.14, 2., this defines a search problem in $\mathrm{PLS}^{\Sigma^{\mathrm{b}}_{i-1}}$. This coincides with the description given by Buss and Krajíček [BK94].

13.6 $\Sigma^{\mathrm{b}}_{i+1}$-definable multi-functions in $\Sigma^{\mathrm{b}}_{i+j}$-$\mathrm{L}^{2+j}\mathrm{IND}$

Let $i \geq 1$, $j \geq 0$, and assume that $\Sigma^{\mathrm{b}}_{i+j}$-$\mathrm{L}^{2+j}\mathrm{IND} \vdash (\forall x)(\exists y)\varphi(x,y)$ with $(\exists y)\varphi(x,y) \in \Sigma^{\mathrm{b}}_{i+1}$, $\varphi \in \Pi^{\mathrm{b}}_i$. By partial cut-elimination we obtain some $\mathrm{BA}^\star$-derivation h such that

- $\mathrm{FV}(h) \subseteq \{x\}$,
- $\Gamma(h) = \{(\exists y)\varphi(x,y)\}$,
- Σ^{b}_i-$\mathrm{crk}(h) \leq j+1$, and
- $\mathrm{o}(h(\underline{a}/x)) = O(|a|_{3+j})$.

We define a search problem by stating its parameters:

- $\Phi := \mathrm{deco}(h)$ is a finite set of formulae in BFOR,
- as the "complexity class" we take $\mathcal{C} := \Sigma^{\mathrm{b}}_i$,
- for the size parameter we choose $s := |h|$,
- the initial value function is given by $h_a := \underbrace{\mathsf{E}\ldots\mathsf{E}}_{j \text{ times}} h(\underline{a}/x)$,
- the formula is as given, $(\exists y)\varphi(x,y)$.

This defines a local search problem according to Definition 13.12, because

- $\Gamma(h_a) = \{(\exists y)\varphi(\underline{a},y)\}$,
- Σ^{b}_i-$\mathrm{crk}(h_a) \leq 1$,
- $\mathrm{o}(h_a) \leq 2_j(\mathrm{o}(h(\underline{a}/x))) = 2_j(\mathcal{O}(|a|_{3+j}))$,
- as $h(\underline{a}/x) \in \mathcal{H}^s_{\mathrm{BA}}$ we have

$$\begin{aligned}
\vartheta(h_a)(s) &= \vartheta(\underbrace{\mathsf{E}\ldots\mathsf{E}}_{j\times} h(\underline{a}/x))(s) \\
&= \mathrm{o}(\underbrace{\mathsf{E}\ldots\mathsf{E}}_{(j-1)\times} h(\underline{a}/x)) \cdot (\vartheta((\underbrace{\mathsf{E}\ldots\mathsf{E}}_{(j-1)\times} h(\underline{a}/x))(s)+2) \\
&= 2_{j-1}(\mathcal{O}(|a|_{3+j})) \cdot (\vartheta((\underbrace{\mathsf{E}\ldots\mathsf{E}}_{(j-1)\times} h(\underline{a}/x))(s)+2) \\
&= \cdots = \mathcal{O}(|a|)
\end{aligned}$$

- $\mathrm{deco}(h_a) \subseteq \Phi_a$ by Lemma 13.8, 1.

As $\mathrm{o}(h_a) = O(||a||)$, Proposition 13.14, 3., shows that the computed multi-function of this search problem is in $\mathrm{FP}^{\Sigma^{\mathrm{b}}_i}[\mathrm{wit}, 2_j(\mathcal{O}(\log^{2+j} n))]$, which coincides with the description given by Pollett [Pol99].

Acknowledgements

The authors gratefully acknowledge support by the Engineering and Physical Sciences Research Council (EPSRC) under grant number EP/D03809X/1.

References

[AJ05] Klaus Aehlig and Felix Joachimski. Continuous normalization for the lambda-calculus and Gödel's T. *Annals of Pure and Applied Logic*, 133(1–3):39–71, May 2005.

[AS00] Klaus Aehlig and Helmut Schwichtenberg. A syntactical analysis of non-size-increasing polynomial time computation. In *Proceedings of the Fifteenth IEEE Symposium on Logic in Computer Science (LICS '00)*, pages 84 – 91, June 2000.

[Bec01] Arnold Beckmann. Exact bounds for lengths of reductions in typed λ-calculus. *Journal of Symbolic Logic*, 66(3):1277–1285, 2001.

[Bec03] Arnold Beckmann. Dynamic ordinal analysis. *Arch. Math. Logic*, 42:303–334, 2003.

[Bec06] Arnold Beckmann. Generalised dynamic ordinals—universal measures for implicit computational complexity. In *Logic Colloquium '02*, volume 27 of *Lect. Notes Log.*, pages 48–74. Assoc. Symbol. Logic, La Jolla, CA, 2006.

[BK94] Samuel R. Buss and Jan Krajíček. An application of Boolean complexity to separation problems in bounded arithmetic. *Proc. London Math. Soc. (3)*, 69(1):1–21, 1994.

[Buc91] Wilfried Buchholz. Notation systems for infinitary derivations. *Archive for Mathematical Logic*, 30:277–296, 1991.

[Buc97] Wilfried Buchholz. Explaining Gentzen's consistency proof within infinitary proof theory. In *Computational logic and proof theory (Vienna, 1997)*, volume 1289 of *Lecture Notes in Comput. Sci.*, pages 4–17. Springer, Berlin, 1997.

[Bus86] Samuel R. Buss. *Bounded arithmetic*, volume 3 of *Studies in Proof Theory. Lecture Notes.* Bibliopolis, Naples, 1986.

[Bus04] Samuel R. Buss. Bounded arithmetic and constant depth Frege proofs. In *Complexity of computations and proofs*, volume 13 of *Quad. Mat.*, pages 153–174. Dept. Math., Seconda Univ. Napoli, Caserta, 2004.

[Gen35a] Gerhard Gentzen. Untersuchungen über das logische Schließen. I. *Mathematische Zeitschrift*, 39:176–210, 1935.

[Gen35b] Gerhard Gentzen. Untersuchungen über das logische Schließen. II. *Mathematische Zeitschrift*, 39:405–431, 1935.

[Göd58] Kurt Gödel. Über eine bisher noch nicht benützte Erweiterung des finiten Standpunkts. *Dialectica*, 12:280–287, 1958.

[KMS75] G. Kreisel, G.E. Mints, and S.G. Simpson. The use of abstract language in elementary metamathematics: Some pedagogic examples. In R. Parikh, editor, *Logic Colloquium*, volume 453 of *Lecture Notes in Mathematics*, pages 38–131. Springer, 1975.

[Kra93] Jan Krajíček. Fragments of bounded arithmetic and bounded query classes. *Trans. Amer. Math. Soc.*, 338(2):587–598, 1993.

[Min78] Grigori E. Mints. Finite investigations of transfinite derivations. *Journal of Soviet Mathematics*, 10:548–596, 1978. Translated from: Zap. Nauchn. Semin. LOMI 49 (1975). Cited after Grigori Mints. *Selected papers in Proof Theory.*Studies in Proof Theory. Bibliopolis, 1992.

[Pol99] Chris Pollett. Structure and definability in general bounded arithmetic theories. *Ann. Pure Appl. Logic*, 100(1-3):189–245, 1999.

[PW85] J. Paris and A. Wilkie. Counting problems in bounded arithmetic. In A. Dold and B. Eckmann, editors, *Methods in Mathematical Logic (Proceedings Caracas 1983)*, number 1130 in Lecture Notes in Mathematics, pages 317–340. Springer, 1985.

[Sch51] Kurt Schütte. Die unendliche Induktion in der Zahlentheorie. *Mathematische Annalen*, 122:369–389, 1951.

[Tai68] William W. Tait. Normal derivability in classical logic. In J. Barwise, editor, *The Syntax and Semantics of Infinitatry Languages*, number 72 in Lecture Notes in Mathematics, pages 204–236. Springer, 1968.